

LLM Feedback for Topic Model with Preference Optimization and Contrastive Learning

Tung Nguyen^{1,2†}, Dat Phan Trong¹, Linh Ngo Van¹, Duc Anh Nguyen¹,
Hoang Thai Dinh², Diep Nguyen²

¹Hanoi University of Science and Technology (HUST), Vietnam

²University of Technology Sydney (UTS), Australia

Abstract

The integration of large language models (LLMs) into topic modeling has shown promise for improving topic interpretability and coherence, particularly by exploiting the generative capabilities of LLMs through instruction-based prompting. However, existing methods often incur substantial computational cost and inference complexity. To address these limitations, we propose **LLMFeed**, a novel framework that leverages **feedback** from an **LLM** to improve topic model optimization. Specifically, the LLM is used to **evaluate** topic quality, **identify** semantic meaning, and **recommend** representative topic words. These signals are then incorporated to (i) refine topic representations via Direct Preference Optimization (DPO), and (ii) guide contrastive learning for improved document-topic assignments. Extensive experiments show that **LLMFeed** achieves a 7.6% gain in topic coherence and over 10% improvement in doc-topic distributions on 20 Newsgroups dataset, while requiring only minutes to run, compared with hours for recent LLM-integrated topic modeling methods. The source code is available at: <https://github.com/tungserver99/LLMFeed.git>.

1 Introduction

Neural Topic Models (NTMs) (Srivastava and Sutton, 2017; Dieng et al., 2020; Nguyen et al., 2025e; Xuan et al., 2026) have become a dominant approach for uncovering latent themes in document corpora. An effective NTM should produce coherent, diverse topics and accurate document-topic assignments (Wu et al., 2024a). Recent work has integrated large language models (LLMs) (Brown et al., 2020; Grattafiori et al., 2024), typically via contextualized embeddings (BehnamGhader et al., 2024), to improve NTMs performance. However, important limitations remain. In VAE-based NTMs (Bianchi et al., 2021a; Pham et al., 2024b;

Topic Name	Top words	Good words not in Top Words
Concert and Event Attendance	concert, concerts, attend, hall, palace, visited, shortly , near , queen , arrested	attendance, crowds, audience, ticket
Art and Photography	dark, famous , circles, inspired , photography, fashion, strange , art, colour, universe	images, drawing, imaging, present

Table 1: Example of relevant/irrelevant words in topics learned by ECRTM on the BBC dataset (Greene and Cunningham, 2006). Irrelevant words are highlighted in blue. A good refinement method should replace them with more appropriate words outside the top words.

Nguyen et al., 2025e; Pham Khanh et al., 2026b), LLM embeddings mainly improve document-topic alignment with limited gains in topic quality. In contrast, clustering-based approaches (Grootendorst, 2022; Angelov and Inkpen, 2024) often suffer from limited topic diversity and weaker document-topic inference (Wu et al., 2024a).

Beyond embedding-based integration, LLMs bring an explicit semantic judgment capability to topic modeling: they can assess whether a set of topic words forms a coherent, interpretable concept, rather than only optimizing statistical proxy objectives. Recent work has therefore used the generative capabilities of LLMs to construct or refine topic representations. However, when used directly to infer corpus-level themes, as in TopicGPT (Pham et al., 2024a), LLMs often produce broad high-level concepts rather than the fine-grained latent aspects extracted by NTMs (Yang et al., 2025). TopicGPT also requires querying the entire corpus with an LLM, leading to substantial runtime and API costs.

LLM-ITL (Yang et al., 2025) instead refines topic words by filtering LLM suggestions through the dataset vocabulary, combining LLM-based se-

[†]Corresponding author

mantic judgment with the fine-grained extraction ability of NTMs. While it improves coherence over base models such as ECRTM (Wu et al., 2023), it requires high-end GPUs (e.g., an A100 with ~ 80 GB memory) and thousands of LLM calls because the LLM is invoked at every training batch. Moreover, its gains have little impact on document-topic assignments, which remain comparable to the base model (Yang et al., 2025).

To address these limitations, we propose **LLM-Feed**, a framework that efficiently injects **feedback** from an **LLM** into topic modeling. As shown in Table 1, top- m topic words may contain irrelevant terms, while better candidates can lie outside the list. LLMFeed uses LLM feedback to construct word preferences and applies Direct Preference Optimization (DPO) (Rafailov et al., 2023) to suppress noisy terms and promote better alternatives, yielding more coherent topic words and descriptive topic summaries. These summaries are further used to guide document-topic learning: LLMFeed embeds documents and topic descriptions using a pre-trained language embedding model (PLEM), identifies the top- t relevant topics for each document, and applies contrastive learning (Khosla et al., 2020) so that documents sharing relevant topics have closer topic distributions than unrelated ones. Importantly, LLMFeed requires **only a single LLM feedback phase**, whose signals are reused during training to avoid repeated LLM calls. To the best of our knowledge, this is the first work to apply DPO to optimize topic models.

We summarize the contributions as follows:

- For better topic quality, we introduce an **LLM-guided topic refinement** mechanism that constructs word-level preferences from a single LLM feedback phase and applies DPO to optimize topic words.
- For document-topic assignments, we propose a **topic-guided contrastive learning** strategy that uses LLM-generated topic descriptions to align documents with their underlying topics.
- Extensive experiments across multiple datasets show that **LLMFeed** achieves strong performance in both topic quality and document-topic assignment, while maintaining efficient LLM usage.

2 Related works

Topic models. Traditional topic modeling methods mainly follow two directions: probabilistic graphical models (PGMs) (Koller and Friedman, 2009; Hofmann, 1999; Blei et al., 2003; Blei and Lafferty, 2005; Blei, 2012) and non-negative matrix factorization (NMF) (Lee and Seung, 2000; Kim et al., 2015; Shi et al., 2018). Despite their success, these approaches often suffer from limited performance and expensive parameter inference (Wu et al., 2024a). Neural topic models (NTMs) (Srivastava and Sutton, 2017; Pham et al., 2024b; Nguyen et al., 2025c; Pham Khanh et al., 2026a) have therefore emerged as promising alternatives, enabling more efficient training through gradient-based optimization.

Most recent NTMs (Wu et al., 2024a; Vuong et al., 2025; Nguyen et al., 2026c,d) adopt a variational autoencoder (VAE) framework (Kingma and Welling, 2014). Recent work improves the encoder through techniques such as contrastive learning on augmented texts (Nguyen and Luu, 2021) and modeling document-topic relationships with Optimal Transport (OT) distance (Zhao et al., 2021; Wu et al., 2024b). On the decoder side, several methods incorporate pre-trained word embeddings (Dieng et al., 2020; Wu et al., 2023), often with additional constraints such as conditional transport (Wang et al., 2022) or OT distance (Pham et al., 2024b). More recently, graph structures have been explored to improve topic coherence (Adhya and Sanyal, 2024; Liu et al., 2025). Neural topic models have also shown strong performance in specialized settings, including short texts (Nguyen et al., 2025a; Vu et al., 2025), dynamic environments (Nguyen et al., 2026a), and cross-lingual scenarios (Nguyen et al., 2025b; Phat et al., 2026; Nguyen et al., 2026b).

LLMs in topic models. Recent advances in large language models (LLMs) have attracted growing interest in their integration with topic modeling. One prominent direction appears in clustering-based NTMs (Grootendorst, 2022; Angelov and Inkpen, 2024; Ma et al., 2025), which use pre-trained LLM embeddings to cluster documents and then extract representative words from each cluster to form topics. Another line of work incorporates LLM-based embeddings into the encoder of NTMs (Bianchi et al., 2021a,b), or uses them as semantic signals to regularize document-topic distributions (Nguyen et al., 2025e,d). Beyond contextual embeddings, the generative capabilities of LLMs have also been

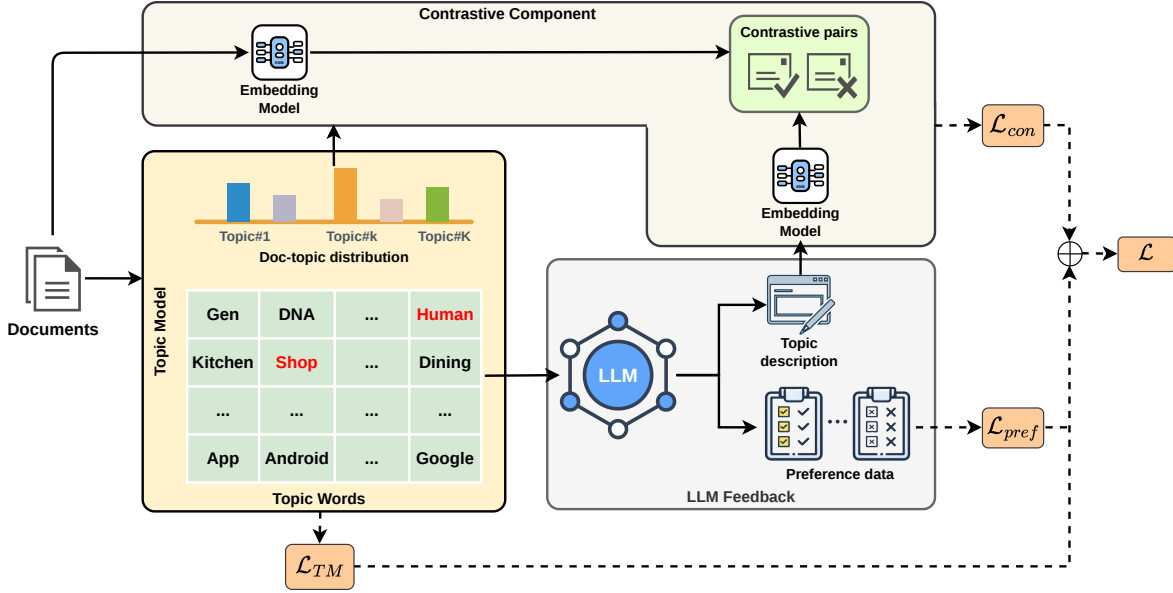


Figure 1: Overview of LLMFeed. A neural topic model generates topic words and document-topic distributions. LLM feedback refines topic quality through preference optimization, while LLM-generated topic descriptions guide contrastive learning to improve document-topic assignments.

exploited to construct topic representations (Mu et al., 2024; Pham et al., 2024a) or refine topic words of existing topic models (Yang et al., 2025). However, these approaches often incur substantial computational costs.

3 Method

This section first introduces the base topic model, followed by the key components of LLMFeed. An overview of our method is illustrated in Figure 1.

3.1 Preliminaries

Following recent neural topic models (Wu et al., 2024a; Nguyen et al., 2025e), we build our model based on the VAE framework. Let $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^D$ denote a corpus of D documents with a vocabulary of size V . Each document $\mathbf{x}_i$ is represented by its Bag-of-Words vector $x_i^{\text{bow}} \in \mathbb{R}^V$. The objective is to learn the topic-word association matrix $\beta = (\beta_1, \dots, \beta_K) \in \mathbb{R}^{V \times K}$ for K topics, together with the topic proportion $\theta_i \in \mathbb{R}^K$ for each document. In the VAE-based NTM, x_i^{bow} is fed into the encoder to infer the parameters of a Gaussian posterior: $\mu_i = h_\mu(x_i^{\text{bow}})$ and $\Sigma_i = \text{diag}(h_\Sigma(x_i^{\text{bow}}))$. A latent variable $\mathbf{z}_i$ is sampled from posterior distribution $q(\mathbf{z}_i | \mathbf{x}_i) = \mathcal{N}(\mathbf{z}_i | \mu_i, \Sigma_i)$, while the prior is specified as $p(\mathbf{z}_i) = \mathcal{N}(\mathbf{z}_i | \mu_0, \Sigma_0)$, with μ_0 and Σ_0 being hyper-parameters. The document-topic proportion is then computed as $\theta_i = \text{softmax}(\mathbf{z}_i)$.

To improve topic coherence, many topic models incorporate pre-trained word embeddings into β . Given embedding dimension L , we define the word embedding matrix $\mathbf{W} = (\mathbf{W}_1, \dots, \mathbf{W}_V) \in \mathbb{R}^{L \times V}$ and the topic embedding matrix $\mathbf{T} = (\mathbf{T}_1, \dots, \mathbf{T}_K) \in \mathbb{R}^{L \times K}$. Some approaches parameterize β as the product of $\mathbf{W}$ and $\mathbf{T}$ (Dieng et al., 2020), but this formulation often reduces topic diversity (Wu et al., 2023). Instead, we adopt the soft-assignment mechanism from ECRTM (Wu et al., 2023), which treats topics as cluster centers in the word embedding space to encourage diverse topics:

$$\beta_{vk} = \frac{\exp(-\|\mathbf{W}_v - \mathbf{T}_k\|^2/\tau)}{\sum_{k'=1}^K \exp(-\|\mathbf{W}_v - \mathbf{T}_{k'}\|^2/\tau)},$$

where the logit $\ell_{vk} = -\frac{\|\mathbf{W}_v - \mathbf{T}_k\|^2}{\tau}$ denotes the unnormalized compatibility score between word embedding $\mathbf{W}_v$ and topic embedding $\mathbf{T}_k$; and τ is a temperature hyper-parameter. The topic modeling objective is formulated as:

$$\mathcal{L}_{\text{TM}} = \frac{1}{D} \sum_{i=1}^D \left[- (x_i^{\text{bow}})^\top \log(\text{softmax}(\beta \theta_i)) + \text{KL}(q(\mathbf{z}_i | \mathbf{x}_i) \| p(\mathbf{z}_i)) \right].$$

Furthermore, the Embedding Clustering Regularization loss $\mathcal{L}_{\text{ECR}}$ (Wu et al., 2023) is applied to further promote topic diversity. Details of $\mathcal{L}_{\text{ECR}}$ are provided in Appendix A.

3.2 LLM-driven Topic Feedback

Following the common topic-modeling protocol, we represent and evaluate each topic by its **top-15** words ranked by word-topic compatibility (Wu et al., 2024a). Since topics from existing NTMs may still contain irrelevant, generic, or ambiguous words, LLMFeed refines them through topic evaluation, meaning identification, and preference word recommendation.

Feedback construction. First, the LLM scores each topic from its top-15 words as incoherent, partially coherent with noise, or strongly coherent (Scores 1-3). Second, the LLM infers a short topic description based on the list of top words; for low-quality topics, we provide a slightly larger word context together with existing topic descriptions, and instruct the LLM to avoid semantic overlap when recovering the dominant meaning.

Finally, LLMFeed detects weak top-15 words and recommends better alternatives from a 30-word candidate pool. To balance the 15 existing topic words, we add 15 outside candidates: words ranked 16–25 by the base NTM and 5 PLEM-retrieved vocabulary words (Chen et al., 2024). The former preserves the learned topic structure, while the latter recovers useful words that the base model may under-rank. For PLEM retrieval, we embed the topic description and vocabulary words in the same semantic space, and select the top-5 most similar words outside the NTM’s top-25 list. Given the pool and topic description, the LLM selects relevant words as “wins” and noisy or generic words as “losses”, while preserving good top-15 words.

Let $\mathcal{V}_k^+$ and $\mathcal{V}_k^-$ denote the win and lose word sets for topic k , we construct the preference dataset as $\mathcal{D}_k^{\text{pref}} = \{(k, v^+, v^-) \mid v^+ \in \mathcal{V}_k^+, v^- \in \mathcal{V}_k^-\}$. Each LLM prompting step in LLMFeed uses a single prompt with function calling (Chen et al., 2025) to ensure structured outputs. We use GPT-4o via the OpenAI API as the LLM. Prompt templates, function schemas and reason for LLM choice are provided in Appendix B.

Why ground LLM judgments in corpus words?

As discussed in Section 1, LLMs provide semantic judgments closer to human interpretation. However, free-form generation is not always aligned with topic modeling, whose goal is to discover coherent, diverse, and fine-grained latent themes characteristic of the given corpus (Wu et al., 2024a). Direct LLM-based generation may produce broad

conceptual summaries rather than detailed corpus-specific topics. LLM-ITL (Yang et al., 2025) filters LLM-suggested words through the corpus vocabulary, but discarded out-of-vocabulary suggestions mean that part of the LLM feedback may be unused. LLMFeed instead grounds the candidate space in the corpus vocabulary from the beginning, and uses the LLM identify relevant or noisy words.

Furthermore, although distinguishing good topic words from noisy ones may appear simple to humans or strong LLMs, it requires semantic understanding that cheap proxy signals often miss. TF-IDF-style filtering and embedding similarity are already used in prior methods (Grootendorst, 2022; Wu et al., 2023), yet noisy or weakly related words can still appear in discovered topics, as shown in Table 1. LLMFeed therefore leverages LLM knowledge as a high-quality semantic signal to guide topic-word preference construction.

3.3 Topic Modeling with LLM Feedback

The generated preference dataset is then used to fine-tune the base topic model via preference optimization. In standard DPO settings for LLMs (Rafailov et al., 2023), the policy corresponds to the probability distribution over outputs. Analogously, in topic modeling, the policy for word v under topic k can be defined as $\pi_\phi(v|k)$, where ϕ denotes the parameters of the NTM. In many NTMs (Srivastava and Sutton, 2017; Dieng et al., 2020), this corresponds to β_{vk} .

However, in our base model ECRTM, β_{vk} represents the soft assignment of word v to topic k induced by the clustering mechanism and is normalized over K topics rather than over the vocabulary, unlike prior approaches such as ETM (Dieng et al., 2020). Therefore, we define the policy as $\pi_\phi(v|k) = \text{softmax}_v(\ell_{:,k})_v$, where ℓ denotes the unnormalized compatibility scores between word and topic embeddings as described in Section 3.1. The reference policy obtained after training the base NTM is written as $\pi_\phi^{\text{ref}}(v|k)$.

The observed preferences are assumed to be generated by an implicit reward function $r(k, v)$. Under the Bradley-Terry (BT) model (Bradley and Terry, 1952), the preference probability is:

$$P(v^+ \succ v^- \mid k) = \sigma(r(k, v^+) - r(k, v^-)).$$

Rather than explicitly learning this reward and then optimizing it with reinforcement learning (RL), we follow Direct Preference Optimization

(DPO) (Rafailov et al., 2023). From an RL perspective, DPO can be viewed as the closed-form preference optimization objective corresponding to KL-regularized reward maximization, avoiding the need to fit an explicit reward model. In our setting, each pair (k, v^+, v^-) indicates that word v^+ is preferred over v^- under topic k , and the reference policy is obtained from the base topic model before LLM-guided refinement, denoted as $\pi_\phi^{\text{ref}}(v|k)$.

This yields the topic-level preference loss:

$$\begin{aligned}\mathcal{L}_{\text{pref}}^k &= -\mathbb{E}_{(k, v^+, v^-)} \log \sigma \left(\alpha \log \frac{\pi_\phi(v^+|k)}{\pi_\phi^{\text{ref}}(v^+|k)} \right. \\ &\quad \left. - \alpha \log \frac{\pi_\phi(v^-|k)}{\pi_\phi^{\text{ref}}(v^-|k)} \right) \\ &= -\mathbb{E}_{(k, v^+, v^-)} \log \sigma \left[\alpha (\ell_{v^+, k} - \ell_{v^-, k}) \right. \\ &\quad \left. - (\ell_{v^+, k}^{\text{ref}} - \ell_{v^-, k}^{\text{ref}}) \right],\end{aligned}\quad (1)$$

where α is the preference margin hyperparameter. Further derivation details of $\mathcal{L}_{\text{pref}}^k$ are provided in Appendix C. During optimization, reference ℓ^{ref} will be frozen. Finally, the overall preference loss is obtained by aggregating over all topics: $\mathcal{L}_{\text{pref}} = \sum_{k=1}^K \mathcal{L}_{\text{pref}}^k$. This formulation adapts DPO to topic modeling by defining the policy over vocabulary words conditioned on topics, rather than token sequences.

3.4 Topic-guided Contrastive Learning

The topic identities inferred by the LLM not only help construct preference data for topic-word relationships but also guide document-topic assignments. For each topic, we form a topic sentence representing its meaning:

Topic about <topic summary> with
top words: <top 15 words>.

These topic sentences and all training documents are embedded using the same pre-trained language embedding model (PLEM). For each document, we then select the top- t topics with the highest similarity scores. Based on these assignments, contrastive data for a document x_i is constructed as follows:

- Positive documents: documents that share at least one top- t topic with x_i .
- Negative documents: documents that do not share any of the top- t topics with x_i .

We use top- t topics rather than only top-1 when constructing the contrastive sets. This improves

robustness for two reasons: similarity scores derived from PLEM embeddings may contain noise, and documents often exhibit mixtures of multiple topics, making a single top topic unreliable.

Using these contrastive sets, we encourage documents with similar topic identities to have similar topic representations. Since each document may have multiple positive samples, we adopt the supervised contrastive (SupCon) loss (Khosla et al., 2020). For an anchor document x_i , the contrastive loss over its topical representation θ_i is defined as:

$$\mathcal{L}_{\text{con}}^i = -\frac{1}{|\mathcal{P}(i)|} \sum_{p \in \mathcal{P}(i)} \log \frac{\exp\left(\frac{s(\theta_i, \theta_p)}{\gamma}\right)}{\sum_{a \neq i} \exp\left(\frac{s(\theta_i, \theta_a)}{\gamma}\right)},$$

where $s(\cdot, \cdot)$ denotes cosine similarity, $\mathcal{P}(i)$ denotes the positive set of x_i , and γ is a temperature parameter. This objective pulls topic representations of documents sharing similar topics closer while pushing apart those of unrelated documents. The overall contrastive loss is obtained by averaging over all documents: $\mathcal{L}_{\text{con}} = \frac{1}{D} \sum_{i=1}^D \mathcal{L}_{\text{con}}^i$.

3.5 Overall objective function

For the LLM Feedback to be effective, the base topic model must first learn stable topic representations. Therefore, LLMFeed adopts two training stages separated by an offline LLM feedback step.

In Stage 1, the base topic model is trained using the standard topic modeling loss together with the ECR loss described in Section 3.1:

$$\mathcal{L}_{\text{stage1}} = \mathcal{L}_{\text{TM}} + \lambda_{\text{ECR}} \mathcal{L}_{\text{ECR}}. \quad (2)$$

Stage 1 runs for E epochs, where E is a hyperparameter. After that, LLM-driven Topic Feedback is performed to generate topic summaries and construct auxiliary artifacts, including preference data and embeddings of topic descriptions and documents. Once generated, these artifacts are fixed and the LLM is no longer required.

Stage 2, which runs for E' epochs, then incorporates preference optimization and contrastive learning. The overall training objective becomes:

$$\begin{aligned}\mathcal{L}_{\text{stage2}} &= \mathcal{L}_{\text{TM}} + \lambda_{\text{ECR}} \mathcal{L}_{\text{ECR}} \\ &\quad + \lambda_{\text{pref}} \mathcal{L}_{\text{pref}} + \lambda_{\text{con}} \mathcal{L}_{\text{con}}.\end{aligned}\quad (3)$$

Here, λ_{ECR} , λ_{pref} , and λ_{con} are weighting hyperparameters. This objective can also be viewed as adding LLM-informed semantic regularization to the base NTM. In the base model, the main inductive biases come from the Gaussian latent prior

Method	20NG				WOS				BBC				NYT			
	C_v	LLMscore	NPMI	TD	C_v	LLMscore	NPMI	TD	C_v	LLMscore	NPMI	TD	C_v	LLMscore	NPMI	TD
LDA	0.393	2.23	-0.031	0.609	0.373	2.58	-0.017	0.671	0.364	2.32	-0.041	0.466	0.390	2.60	-0.029	0.545
ETM	0.399	1.98	-0.024	0.712	0.379	2.44	-0.009	0.793	0.370	2.13	-0.029	0.838	0.388	2.31	-0.021	0.702
CTM	0.323	1.96	-0.085	0.752	0.332	2.63	-0.029	0.872	0.379	2.53	-0.023	0.745	0.322	2.14	-0.077	0.884
ECRTM	0.477	2.57	0.039	0.915	0.434	2.68	0.028	0.960	0.440	2.81	0.025	0.922	0.422	2.73	0.022	0.979
Bertopic	0.370	2.29	-0.038	0.480	0.409	2.69	0.001	0.555	0.430	2.80	-0.003	0.689	0.391	2.66	-0.019	0.510
Ctop2vec	0.450	2.53	0.023	0.532	0.412	2.69	-0.016	0.628	0.386	2.75	-0.015	0.425	0.339	2.56	-0.022	0.486
CAST	0.392	2.42	-0.020	0.478	0.423	2.69	-0.002	0.612	0.370	2.76	-0.007	0.483	0.327	2.64	-0.021	0.534
FASTopic	0.410	2.18	-0.033	<u>0.927</u>	0.407	2.71	-0.004	0.964	0.340	2.25	-0.049	0.962	0.391	2.55	-0.031	0.971
Neuromax	0.491	2.63	0.046	0.917	0.441	2.69	0.030	0.965	0.444	2.82	0.026	0.921	0.426	2.77	0.025	0.984
CGTM	0.422	2.33	-0.021	0.719	0.444	2.72	0.021	0.702	0.438	2.61	-0.006	0.647	<u>0.432</u>	2.65	-0.001	0.790
LLM-ITL	<u>0.497</u>	2.63	0.048	0.859	<u>0.455</u>	<u>2.75</u>	<u>0.039</u>	0.958	<u>0.451</u>	2.84	0.031	0.885	0.430	<u>2.79</u>	<u>0.028</u>	0.972
LLMFeed	0.514	2.74	0.057	0.928	0.466	2.82	0.046	0.977	0.466	2.91	0.039	<u>0.931</u>	0.448	2.89	0.037	0.981

Table 2: Comparison of topic quality across four datasets using four metrics, averaged over num Topics $K = 25, 50, 75, 100$. Best and second-best results are shown in bold and underline.

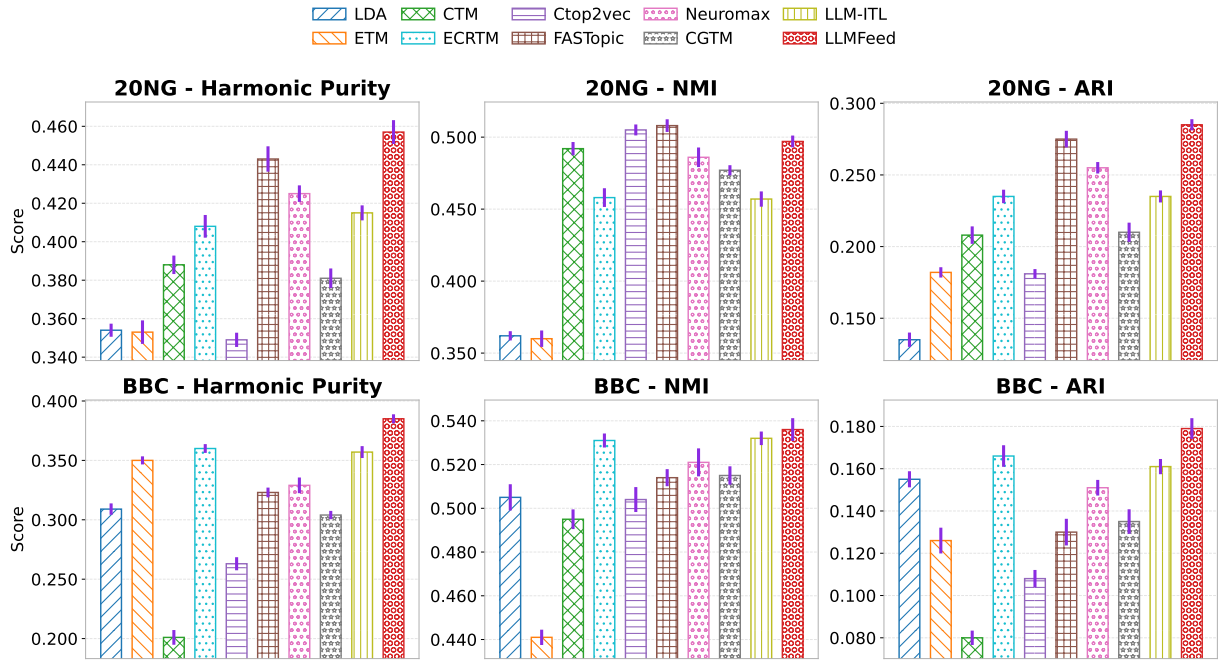


Figure 2: Document-topic assignment results measured by Harmonic Purity, NMI and ARI on 20NG and BBC datasets. Results are mean of five runs; purple bars indicate std.

and word-embedding-based topic structure. LLM-Feed complements these signals with higher-level semantic guidance: the preference loss biases topic-word distributions toward LLM-preferred corpus words, while the contrastive loss biases document-topic representations toward LLM-described topic meanings. Thus, LLM feedback acts as an additional semantic regularizer rather than changing the original VAE prior.

4 Experiment

4.1 Experiment settings

Datasets and Metrics. We evaluate our method on four benchmark datasets: **20 News-groups (20NG)** (Lang, 1995), **Web of Science (WoS)** (Kowsari et al., 2017), **BBC** (Greene

and Cunningham, 2006), and **New York Times (NYT)** (Wu et al., 2024b). We assess both *topic quality* and *document-topic assignments*. Topic quality is measured using C_v and NPMI coherence (with the Wikipedia reference corpus¹); Topic Diversity (TD) (Wu et al., 2023); and the LLM score (Stammach et al., 2023). Document-topic assignment quality is evaluated with Harmonic Purity, NMI, and ARI. Higher values indicate better performance.

Baseline Models. We compare our method with LDA (Blei et al., 2003), ETM (Dieng et al., 2020), CTM (Bianchi et al., 2021a), ECRTM (Wu et al., 2023), Bertopic (Grootendorst, 2022),

¹<https://github.com/dice-group/Palmetto/>

Model	LLMscore	Topic description
TopicGPT	-	Politics: Mentions libertarian and anarchist ideologies.
	-	Technology: Relates to the application of scientific knowledge for practical purposes.
ECRTM	2	• adults sex homosexuals homosexual sexual gay discrimination male population adult minorities homosexuality handguns percentage firearm
	2	• input byte file output bytes files stream sender log password fails algorithm entry corresponding null
	2	• encryption cryptographic encrypted cryptography wiretap communication secrets chip secret computing escrow pgp trusted network keys
	3	• linear probability parameters measured geometry density variable corresponding quantum gravity computation differential radius infinite mathematical
LLM-ITL	3	• Social Injustice: prejudice discrimination ignorance bias racism homosexuality homosexuals minorities racial gay prejudice bias racism sexual sex tolerance racist orientation
	3	• Data conversion: binary format byte sequence output input algorithm stream code convert file bytes generate corresponding compiler
	3	• Data Security: cryptography encrypted protected cryptographic secure reliable access key trusted password crucial secrets gateway chip sensitive
	2	• Cryptography Algorithms: cryptography algorithms encryption quantum logic computation electronic circuits computing software computer mathematical networking cryptographic theoretical
LLMFeed	3	• Homosexuality and Discrimination: homosexuals homosexual homosexuality discrimination minorities racial gay prejudice bias racism sexual sex tolerance racist orientation
	3	• Data Input and Output Operations: input byte output file bytes files stream entry log bits processing pointer inputs pin interrupt
	3	• Cryptographic Encryption and Privacy: encryption cryptographic encrypted cryptography password trusted escrow pgp protocol secret secure security key keys privacy
	3	• Mathematical and Computational Theory: mathematical quantum geometry linear infinite computation theory probability theoretical differential hypothesis implies algorithm gravity algorithms

Table 3: Topic descriptions and LLMscore produced by different models on 20NG datasets. Words highlighted in blue denote terms that are weakly related to the corresponding topic, while words highlighted in red indicate terms that appear in multiple topics.

Ctop2vec (Angelov and Inkpen, 2024), CAST (Ma et al., 2025), FASTopic (Wu et al., 2024b), Neuromax (Pham et al., 2024b), CGTM (Liu et al., 2025), and LLM-ITL (Yang et al., 2025). Since TopicGPT (Pham et al., 2024a) does not produce topic words, we follow prior work and compare its topics qualitatively (Yang et al., 2025).

Due to space limitations, the main paper reports the key results, while the Appendices provides experimental details, complete performance tables, additional quantitative results for TopicGPT under a low-topic document-topic assignment setting, sensitivity analysis, feedback design ablations, and comparisons with simpler refinement baselines.

4.2 Topic quality

Overall, Table 2 shows that LLMFeed outperforms other methods, particularly in topic coherence (high C_v and NPMI) and interpretability (high LLM score). Its preference learning mechanism improves topic words quality by replacing low-quality words with more suitable ones. LLM-ITL achieves the second-best performance due to its effective

LLM-in-the-loop design. Among the remaining methods, those leveraging pre-trained word embeddings or graph information - such as ETM, ECRTM, FASTopic, Neuromax, and CGTM - generally produce more coherent topics. In contrast, clustering-based approaches (Bertopic, Ctop2vec, and CAST) do not outperform other models despite using PLEM, suggesting limitations in their clustering mechanisms.

LLMFeed also maintains competitive topic diversity. Its topic meaning identification step explicitly avoids semantic overlap, ensuring more distinct topics. Methods using ECR loss (ECRTM, FASTopic, and Neuromax) also achieve high diversity. While LLM-ITL improves coherence through LLM refinement, it reduces diversity because its topic summarization step does not enforce semantic distinctions between topics. Additional detailed results on topic quality are provided in Appendix E.1.

4.3 Document-topic assignment

Figure 2 presents document-topic assignment results on 20NG and BBC with 100 topics. Bertopic

Model	Minimum GPU	Memory	Time	LLM calls
ECRTM	1 Standard GPU	~1GB	~15m	None
LLM-ITL	A100 (80GB)	~65GB	~8h	2850
TopicGPT	None	None	~23h	11314
Neuromax	1 Standard GPU	~2GB	~20m	None
LLMFeed	1 Standard GPU	~2GB	~23m	300

Table 4: Resource cost on 20NG with 100 topics. “~” denotes approximate values; **red** marks substantially higher memory, runtime, or LLM-calls.

and CAST are excluded because they do not produce document-topic distributions. Overall, LLMFeed achieves the best results on most metrics, benefiting from its topic-guided contrastive learning. The only exception is NMI on 20NG, which is a metric-specific case: LLMFeed still improves Harmonic Purity and ARI on the same dataset. Since NMI emphasizes agreement with the global label partition, embedding- or clustering-oriented methods may better match the broad taxonomy of 20NG, whose labels include closely related groups such as *comp.**, *rec.**, *sci.**, and *talk.**. By contrast, Harmonic Purity and ARI better reflect assignment purity and pairwise consistency, where LLMFeed remains stronger. Additional results are provided in Appendix E.2.

4.4 Topic Representation Comparison

We further inspect the learned topic representations through qualitative visualization. Following the protocol of Adhya et al. (2023), which is also adopted in recent topic-modeling studies (Liu et al., 2025), we align topics across models using a two-step matching strategy for a fair comparison. Table 3 then presents representative matched topics, allowing direct inspection of interpretability, noisy words, and semantic overlap; the reported LLM-score provides an additional quantitative reference.

Overall, TopicGPT produces readable but high-level themes, whereas ECRTM captures more specific topics but includes unrelated words. LLM-ITL improves coherence, yet still shows semantic overlap, such as **Data Security** and **Cryptography Algorithms**. In contrast, LLMFeed reduces overlap through topic meaning identification and preserves stable semantics via base-policy-regularized preference learning, yielding more interpretable and better-separated topics. More quantitative results for TopicGPT are provided in Appendix E.3.

Dataset	Method	C_v	TD	HarmPurity	ARI
20NG	ECRTM	0.485	0.944	0.471	0.297
	DPO-only	<u>0.517</u> (↑6.6%)	<u>0.946</u> (↑0.2%)	0.480 (↑1.9%)	0.304 (↑2.4%)
	Contrastive-only	0.494 (↑1.9%)	<u>0.946</u> (↑0.2%)	0.537 (↑14.0%)	0.363 (↑22.2%)
	LLMFeed	0.522 (↑7.6%)	0.973 (↑3.1%)	<u>0.525</u> (↑11.5%)	<u>0.339</u> (↑14.1%)
WOS	ECRTM	0.450	<u>0.984</u>	0.511	<u>0.324</u>
	DPO-only	<u>0.477</u> (↑6.0%)	0.989 (↑0.5%)	0.506 (↓1.0%)	0.312 (↓3.7%)
	Contrastive-only	0.459 (↑2.0%)	0.989 (↑0.5%)	<u>0.539</u> (↑5.5%)	0.357 (↑10.2%)
	LLMFeed	0.478 (↑6.2%)	0.989 (↑0.5%)	0.546 (↑6.8%)	0.357 (↑10.2%)

Table 5: Ablation study on 20NG and WOS datasets.

4.5 Computational Cost

Table 4 reports the computational cost of training on the 20NG dataset with 100 topics. LLM-ITL and TopicGPT require substantially higher runtime and LLM calls. In contrast, Neuromax completes training within minutes and is only slightly more expensive than ECRTM, mainly due to its OT regularization. LLMFeed also runs within minutes, but still requires 300 LLM calls because it uses three prompts per topic, with additional overhead from $\mathcal{L}_{\text{pref}}$ and $\mathcal{L}_{\text{con}}$. A more in-depth qualitative analysis of the computational cost is presented in Appendix E.4.

4.6 Ablation study

Table 5 presents the ablation results on 20NG and WOS with $K = 50$. The DPO-only variant mainly improves topic quality, yielding clear gains in C_v , but it does not directly optimize document-topic assignments. Thus, its slight decrease in Harmonic Purity and ARI on WOS does not contradict its intended role, especially when categories are semantically close. By contrast, the contrastive-only variant directly improves document-topic relationships, but contributes less to topic quality than the DPO-only and the full model.

Combining both components gives **LLMFeed** the best overall balance. This indicates that the two losses interact rather than simply adding independent gains: DPO improves the semantic quality of topic anchors, which makes the contrastive document-topic supervision more reliable. Therefore, although DPO-only may not always improve document-topic metrics alone, it still contributes positively when paired with contrastive learning.

5 Conclusion

We introduced LLMFeed, a topic modeling framework that incorporates LLM feedback to improve topic quality and document-topic distributions. Our approach combined LLM-guided topic refinement with preference optimization and topic-guided contrastive learning while requiring only a single LLM interaction phase. Experiments on benchmark datasets showed consistent improvements over strong baselines.

Limitations

While LLMFeed improves performance and training efficiency, some limitations remain. First, similar to most topic modeling approaches, LLMFeed requires the number of topics to be specified in advance. Automatically determining an appropriate topic granularity remains an open problem. Future work could explore using LLMs to infer topic granularity directly from the corpus, improving model adaptability. Second, LLMFeed grounds LLM feedback in corpus words to better align with the original goal of topic modeling. However, this also limits the expressiveness of topic representations. A broader future direction is to extend LLMFeed with richer topic descriptions or external linguistic knowledge, while still preserving the corpus-specific semantics of the discovered topics. Third, LLMFeed follows a multi-objective learning formulation, where different training signals may emphasize different aspects of topic modeling. Similar to other recent topic models such as Neuromax (Pham et al., 2024b) and CGTM (Liu et al., 2025), a more principled treatment of objective balancing could further improve the trade-off between topic quality and document-topic assignment. We leave this direction for future work.

Ethical Considerations

We comply with the ACL Code of Ethics and all relevant license terms. Our research in topic modeling is designed to enhance the field. When applied responsibly, it carries no significant societal risks.

Acknowledgments

Linh Ngo Van, Duc Anh Nguyen are funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.05-2025.16.

References

- Suman Adhya, Avishek Lahiri, and Debarshi Kumar Sanyal. 2023. [Do neural topic models really need dropout? analysis of the effect of dropout in topic modeling](#). In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2220–2229, Dubrovnik, Croatia. Association for Computational Linguistics.
- Suman Adhya and Debarshi Kumar Sanyal. 2024. [GINopic: Topic modeling with graph isomorphism network](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6171–6183, Mexico City, Mexico. Association for Computational Linguistics.
- Dimo Angelov and Diana Inkpen. 2024. [Topic modeling: Contextual token embeddings are all you need](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13528–13539, Miami, Florida, USA. Association for Computational Linguistics.
- Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. [Llm2vec: Large language models are secretly powerful text encoders](#). *Preprint*, arXiv:2404.05961.
- Federico Bianchi, Silvia Terragni, and Dirk Hovy. 2021a. [Pre-training is a hot topic: Contextualized document embeddings improve topic coherence](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 759–766, Online. Association for Computational Linguistics.
- Federico Bianchi, Silvia Terragni, Dirk Hovy, Debora Nozza, and Elisabetta Fersini. 2021b. [Cross-lingual contextualized topic models with zero-shot learning](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1676–1683, Online. Association for Computational Linguistics.
- David M. Blei. 2012. [Probabilistic topic models](#). *Commun. ACM*, 55(4):77–84.
- David M. Blei and John D. Lafferty. 2005. [Correlated topic models](#). In *Advances in Neural Information Processing Systems*, volume 18. MIT Press.
- David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. [Latent dirichlet allocation](#). *J. Mach. Learn. Res.*, 3:993–1022.
- Ralph Allan Bradley and Milton E. Terry. 1952. [Rank analysis of incomplete block designs: I. the method of paired comparisons](#). *Biometrika*, 39(3/4):324–345.

- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. [M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.
- Yi-Chang Chen, Po-Chun Hsu, Chan-Jan Hsu, and Dashan Shiu. 2025. [Enhancing function-calling capabilities in LLMs: Strategies for prompt formats, data integration, and multilingual translation](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, pages 99–111, Albuquerque, New Mexico. Association for Computational Linguistics.
- Marco Cuturi. 2013. [Sinkhorn distances: Lightspeed computation of optimal transport](#). In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.
- Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2020. [Topic modeling in embedding spaces](#). *Transactions of the Association for Computational Linguistics*, 8:439–453.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Derek Greene and Padraig Cunningham. 2006. [Practical solutions to the problem of diagonal dominance in kernel document clustering](#). In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006)*, Pittsburgh, Pennsylvania, USA, June 25–29, 2006, volume 148 of *ACM International Conference Proceeding Series*, pages 377–384. ACM.
- Maarten Grootendorst. 2022. [Bertopic: Neural topic modeling with a class-based tf-idf procedure](#). *Preprint*, arXiv:2203.05794.
- Thomas Hofmann. 1999. [Probabilistic latent semantic indexing](#). In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '99*, pages 50–57, New York, NY, USA. Association for Computing Machinery.
- Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. [Supervised contrastive learning](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc.
- Hannah Kim, Jaegul Choo, Jingu Kim, Chandan K. Reddy, and Haesun Park. 2015. [Simultaneous discovery of common and discriminative topics via joint nonnegative matrix factorization](#). In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '15*, pages 567–576, New York, NY, USA. Association for Computing Machinery.
- Diederik P. Kingma and Max Welling. 2014. [Auto-encoding variational bayes](#). In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14–16, 2014, Conference Track Proceedings*.
- Daphne Koller and Nir Friedman. 2009. [Probabilistic Graphical Models: Principles and Techniques](#). Adaptive Computation and Machine Learning. MIT Press.
- Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S. Gerber, and Laura E. Barnes. 2017. [Hdltext: Hierarchical deep learning for text classification](#). In *16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017, Cancun, Mexico, December 18–21, 2017*, pages 364–371. IEEE.
- Ken Lang. 1995. [Newsweeder: Learning to filter net-news](#). In *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9–12, 1995*, pages 331–339. Morgan Kaufmann.
- Daniel Lee and H. Sebastian Seung. 2000. [Algorithms for non-negative matrix factorization](#). In *Advances in Neural Information Processing Systems*, volume 13. MIT Press.
- Jiyuan Liu, Jiaying Yan, Chunjiang Zhu, Xingyu Liu, Li Qing, and Yanghui Rao. 2025. [Neural topic modeling via contextual and graph information fusion](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 13247–13263, Suzhou, China. Association for Computational Linguistics.
- Yanan Ma, Chenghao Xiao, Chenhan Yuan, Sabine N Van Der Veer, Lamiece Hassan, Chenghua Lin, and Goran Nenadic. 2025. [CAST: Corpus-aware self-similarity enhanced topic modelling](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7548–7561, Albuquerque, New Mexico. Association for Computational Linguistics.

- Yida Mu, Chun Dong, Kalina Bontcheva, and Xingyi Song. 2024. [Large language models offer an alternative to the traditional approach of topic modelling](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10160–10171, Torino, Italia. ELRA and ICCL.
- Quang Duc Nguyen, Tung Nguyen, Duc Anh Nguyen, Linh Ngo Van, Sang Dinh, and Thien Huu Nguyen. 2025a. [GloCOM: A short text neural topic model via global clustering context](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1109–1124, Albuquerque, New Mexico. Association for Computational Linguistics.
- Thanh Vinh Nguyen, Ngo Van Dong, Minh Chu Xuan, Tung Nguyen, Linh Ngo Van, Dinh Viet Sang, and Trung Le. 2026a. [Beyond coherence: Improving temporal consistency and interpretability in dynamic topic models](#). In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 3609–3629, Rabat, Morocco. Association for Computational Linguistics.
- Thong Nguyen and Anh Tuan Luu. 2021. [Contrastive learning for neural topic model](#). In *Advances in Neural Information Processing Systems*, volume 34, pages 11974–11986. Curran Associates, Inc.
- Tien Phat Nguyen, Ngo Vu Minh, Tung Nguyen, Linh Ngo Van, Duc Anh Nguyen, Dinh Viet Sang, and Trung Le. 2025b. [XTRA: Cross-lingual topic modeling with topic and representation alignments](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5561–5575, Suzhou, China. Association for Computational Linguistics.
- Tien-Phat Nguyen, Giang Pham Thi Phuong, Thuan Do Phan, Anh-Cuong Le, and Linh Ngo Van. 2026b. [A framework for cross-lingual topic modeling using global context spaces and unified decoding with semantic knowledge distillation](#). *Neurocomputing*, 698:134278.
- Tung Nguyen, Duy-Tung Doan, Huu-Thanh Luong, and Linh Ngo Van. 2026c. [Sub-document neural topic model](#). *Knowledge-Based Systems*, 331:114762.
- Tung Nguyen, Tue Le, Hoang Tran Vuong, Quang Duc Nguyen, Duc Anh Nguyen, Linh Ngo Van, Sang Dinh, and Thien Huu Nguyen. 2025c. [Sharpness-aware minimization for topic models with high-quality document representations](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4507–4524, Albuquerque, New Mexico. Association for Computational Linguistics.
- Tung Nguyen, Linh Ngo Van, Anh Nguyen Duc, and Sang Dinh Viet. 2026d. [Global and local context in short text neural topic model](#). *Artificial Intelligence*, 353:104502.
- Tung Nguyen, Duy-Tung Pham, Quang Duc Nguyen, Linh Ngo Van, Anh Nguyen Duc, and Sang Dinh Viet. 2025d. [Topicot: Neural topic model aligning with pre-trained clustering and optimal transport](#). *Neurocomputing*, 654:131268.
- Tung Nguyen, Linh Ngo Van, Anh Nguyen Duc, and Sang Dinh Viet. 2025e. [A framework for neural topic modeling with mutual information and group regularization](#). *Neurocomputing*, 645:130420.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024a. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.
- OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, and 400 others. 2024b. [Gpt-4o system card](#). *Preprint*, arXiv:2410.21276.
- Chau Minh Pham, Alexander Hoyle, Simeng Sun, Philip Resnik, and Mohit Iyyer. 2024a. [TopicGPT: A prompt-based topic modeling framework](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2956–2984, Mexico City, Mexico. Association for Computational Linguistics.
- Duy-Tung Pham, Thien Trang Nguyen Vu, Tung Nguyen, Linh Van Ngo, Duc Anh Nguyen, and Thien Huu Nguyen. 2024b. [NeuroMax: Enhancing neural topic modeling via maximizing mutual information and group topic regularization](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7758–7772, Miami, Florida, USA. Association for Computational Linguistics.
- Chi Pham Khanh, Tue Le, Anh-Cuong Le, and Linh Ngo Van. 2026a. [A multi-objective surrogate framework for training neural topic models](#). *Neurocomputing*, 682:133432.
- Chi Pham Khanh, Hoang Tran Vuong, Sang Dinh Viet, and Linh Ngo Van. 2026b. [A framework for neural topic modeling using hierarchical clustering and contrastive learning with optimal transport](#). *Neurocomputing*, 670:132530.
- Nguyen Tien Phat, Ngo Vu Minh, Linh Ngo Van, Nguyen Thi Ngoc Diep, and Thien Huu Nguyen. 2026. [Gloctm: Cross-lingual topic modeling via a global context space](#). In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence*,

- Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 32710–32718. AAAI Press.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Tian Shi, Kyeongpil Kang, Jaegul Choo, and Chandan K. Reddy. 2018. [Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations](#). In *Proceedings of the 2018 World Wide Web Conference, WWW '18*, pages 1105–1114, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.
- Akash Srivastava and Charles Sutton. 2017. [Autoencoding variational inference for topic models](#). In *International Conference on Learning Representations*.
- Dominik Stammach, Vilém Zouhar, Alexander Hoyle, Mrinmaya Sachan, and Elliott Ash. 2023. [Revisiting automated topic model evaluation with large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9348–9357, Singapore. Association for Computational Linguistics.
- Tu Vu, Manh Do, Tung Nguyen, Linh Ngo Van, Sang Dinh, and Thien Huu Nguyen. 2025. [Topic modeling for short texts via optimal transport-based clustering](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7666–7680, Vienna, Austria. Association for Computational Linguistics.
- Hoang Tran Vuong, Tue Le, Tu Vu, Tung Nguyen, Linh Ngo Van, Sang Dinh, and Thien Huu Nguyen. 2025. [HiCOT: Improving neural topic models via optimal transport and contrastive learning](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13894–13920, Vienna, Austria. Association for Computational Linguistics.
- Dongsheng Wang, Dandan Guo, He Zhao, Huangjie Zheng, Korawat Tanwisuth, Bo Chen, and Mingyuan Zhou. 2022. [Representing mixtures of word embeddings with mixtures of topic embeddings](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Xiaobao Wu, Xinshuai Dong, Thong Thanh Nguyen, and Anh Tuan Luu. 2023. [Effective neural topic modeling with embedding clustering regularization](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 37335–37357. PMLR.
- Xiaobao Wu, Thong Nguyen, and Anh Tuan Luu. 2024a. [A survey on neural topic models: methods, applications, and challenges](#). *Artif. Intell. Rev.*, 57(2):18.
- Xiaobao Wu, Thong Nguyen, Delvin Ce Zhang, William Yang Wang, and Anh Tuan Luu. 2024b. [Fastopic: Pretrained transformer is a fast, adaptive, stable, and transferable topic model](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 84447–84481. Curran Associates, Inc.
- Minh Chu Xuan, Tien-Phat Nguyen, Linh Ngo Van, Dinh Viet Sang, Nguyen Thi Ngoc Diep, and Trung Le. 2026. [LLM-XTM: Enhancing cross-lingual topic models with large language models](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3719–3737, San Diego, California, United States. Association for Computational Linguistics.
- Xiaohao Yang, He Zhao, Weijie Xu, Yuanyuan Qi, Jueqing Lu, Dinh Phung, and Lan Du. 2025. [Neural topic modeling with large language models in the loop](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1377–1401, Vienna, Austria. Association for Computational Linguistics.
- He Zhao, Dinh Phung, Viet Huynh, Trung Le, and Wray L. Buntine. 2021. [Neural topic model via optimal transport](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Step	Type	Content
Topic evaluation	System	You are a topic modeling evaluator.
Topic evaluation	User	Score topic coherence based on top words. Score 1: unrelated words. Score 2: somewhat related with noise. Score 3: strongly coherent. Topic index: {k}. Top words: {words}.
Topic evaluation	Schema	{ "name": "score_topic", "description": "Score topic coherence from top words", "parameters": { "type": "object", "properties": { "k": { "type": "integer" }, "llm_score": { "type": "integer", "enum": [1,2,3] } }, "required": ["k", "llm_score"] } }.

Table 6: Prompt and function schema for Topic evaluation step.

Step	Type	Content
Topic meaning identification	System	You are a topic modeling assistant.
Topic meaning identification	User (score $\neq 1$)	Generate a short descriptive phrase from top words. Use the dominant semantic theme from top words, prioritizing higher-ranked words. Prefer a concrete noun phrase (2–6 words). Topic index: {k}. Top words: {words}.
Topic meaning identification	User (score = 1)	Generate a short descriptive phrase from top words. Use the dominant semantic theme from top-20 words, prioritizing higher-ranked words. Prefer a concrete noun phrase (2–6 words). Avoid overlap with existing topic descriptions and avoid reusing head nouns. Existing topic descriptions: {used}. Topic index: {k}. Top words (20): {words}.
Topic meaning identification	Schema	{ "name": "describe_topic", "description": "Describe the main meaning of a topic", "parameters": { "type": "object", "properties": { "k": { "type": "integer" }, "main_meaning": { "type": "string" } }, "required": ["k", "main_meaning"] } }.

Table 7: Prompts and function schema for Topic meaning identification step.

A Embedding Clustering Regularization

Embedding Clustering Regularization (ECR) is an effective technique for mitigating the topic collapsing phenomenon frequently observed in neural topic models. The core idea of ECR is to encourage topic embeddings to form well-separated clusters with respect to word embeddings. Following prior work (Wu et al., 2023; Nguyen et al., 2025e), the

ECR objective is defined as:

$$\mathcal{L}_{\text{ECR}} = \sum_{v=1}^V \sum_{k=1}^K \|\mathbf{W}_v - \mathbf{T}_k\|^2 \epsilon_{vk}^* \quad (4)$$

where ϵ^* denotes the optimal transport plan between the set of word embeddings and topic embeddings. Specifically, the empirical distributions of

Step	Type	Content
Preference word recommendation	System	You are a topic preference labeling assistant.
Preference word recommendation	User	Identify win/lose word indices for the topic. IMPORTANT: The indices you return must be the vocabulary indices provided in the word:index pairs below. Do NOT use the position in the list. Only use the numeric values shown after each word. Goal: keep good words in top-15, remove bad words in top-15, and promote good words outside top-15. Definition: - Good (win): semantically central and clearly related to the topic description. - Bad (lose): unrelated, off-topic, too generic, or misleading for the topic. Example: If the topic is about 'computer hardware' and the top-15 contains ['cpu','motherboard','bios','floppy','drive','sale','discount','today',...], good words in top-15 like 'cpu','motherboard' should be win; bad words inside top-15 like 'sale','discount','today' should be lose; good words outside top-15 (e.g., 'chipset') should be win. Index example: if you see [{'cpu': 120}, {'gpu': 450}], then valid indices are 120 and 450 (not 0 or 1). Rules: 1) Any bad words in top15 must be in lose list. 2) Any good words outside top15 must be in win list. 3) Good words inside top15 should be kept as win (unless clearly bad). 4) Aim for a roughly balanced set; if possible keep win/lose counts close (difference ≤ 2). Topic index: {k} Topic description: {descriptions.get(k, "")} Top-15 words: {top15_kv} Other words (rank 16-25 + extra): {other_kv}.
Preference word recommendation	Schema	<pre>{ "name": "build_preferences", "description": "Identify win/lose word indices for a topic", "parameters": { "type": "object", "properties": { "k": { "type": "integer" }, "w_win_indices": { "type": "array", "items": { "type": "integer" } }, "w_lose_indices": { "type": "array", "items": { "type": "integer" } } }, "required": ["k", "w_win_indices", "w_lose_indices"] } }</pre>

Table 8: Prompt and function schema for Preference word recommendation step.

words and topics are represented as $\sum_{v=1}^V \frac{1}{V} \delta_{\mathbf{w}_v}$ and $\sum_{k=1}^K \frac{1}{K} \delta_{\mathbf{T}_k}$, respectively, where δ_x denotes a Dirac mass centered at x . The optimal transport matrix ϵ^* is obtained by solving the following entropy-regularized optimal transport problem:

$$\begin{aligned}
& \text{minimize} && \langle C_{\mathbf{WT}}, \epsilon \rangle - \nu H(\epsilon) \\
& \text{s.t.} && \epsilon \in \mathbb{R}^{V \times K} \\
& && \epsilon \mathbf{1}_K = \frac{1}{V} \mathbf{1}_V, \quad \epsilon^T \mathbf{1}_V = \frac{1}{K} \mathbf{1}_K
\end{aligned} \tag{5}$$

where $C_{\mathbf{WT}} \in \mathbb{R}^{V \times K}$ denotes the pairwise distance matrix between word and topic embeddings. The optimal transport plan ϵ^* is computed using the Sinkhorn algorithm (Cuturi, 2013).

B Details of prompts

This section presents the prompts and function schemas used in our LLM-based preference construction pipeline, including topic evaluation (Table 6), topic meaning identification (Table 7), and preference word recommendation (Table 8).

In our framework, the outputs from each LLM stage serve as inputs for the subsequent stage. This dependency is particularly critical in the preference word recommendation step, where the generated preferences are later used for training with Direct Preference Optimization (DPO). To ensure reliable and well-structured outputs, we employ both chain-of-thought prompting and function calling techniques (Chen et al., 2025), implemented

Algorithm 1 Learning LLMFeed

Require: Document collection $\mathbf{X}$, pretrained word embeddings $\mathbf{W}$, number of topics K , total epochs $N = E + E'$, update start epoch E

Ensure: Topic-word distribution β , document-topic distribution θ

```
1: for  $t = 1, 2, \dots, N$  do
2:   if  $t < E$  then
3:     // Stage 1: base training
4:     Update model parameters using
                                      $\mathcal{L}_{\text{stage1}} = \mathcal{L}_{\text{TM}} + \lambda_{\text{ECR}} \mathcal{L}_{\text{ECR}}.$ 
5:   else if  $t = E$  then
6:     // LLM feedback generation
7:     Save the current model snapshot and the top-10/15/20/25 words of each topic
8:     Evaluate topic quality and generate a short description for each topic
9:     Build topic-level preference labels from top words and embedding-selected extra words
10:    Construct topic sentences from topic descriptions and top words
11:    Encode topic sentences and training documents in a shared embedding space
12:    Assign each document a multi-hot label over its top- $k$  most similar topics
13:   else
14:     // Stage 2: preference and contrastive update
15:     Build the contrastive set for the current batch from the document multi-hot labels
16:     Update model parameters using
```

$$\mathcal{L}_{\text{stage3}} = \mathcal{L}_{\text{TM}} + \lambda_{\text{ECR}} \mathcal{L}_{\text{ECR}} + \lambda_{\text{pref}} \mathcal{L}_{\text{pref}} + \lambda_{\text{con}} \mathcal{L}_{\text{con}}.$$

```
17:   end if
18: end for
```

using the GPT-4o model from OpenAI (OpenAI et al., 2024b).

We choose GPT-4o for three reasons. First, it provides strong semantic judgment quality, which is important for evaluating topic coherence and selecting reliable word preferences. Second, its function-calling ability helps produce structured feedback that can be directly reused by later training stages. Third, it offers a practical cost-quality trade-off: LLMFeed only requires $3K$ LLM calls, where K is the number of topics (typically at most 100), and each prompt mainly contains short word lists. Although open-source LLMs with strong function calling, such as LLaMA-3.1-8B (Grattafiori et al., 2024), could also be used, deploying large local models may require additional high-end hardware. Using GPT-4o allows LLMFeed to run with a standard GPU plus a small API cost.

C DPO loss details

Recall that in our setting, preferences are defined over words within a topic. Under the Bradley-Terry model, the probability that word v^+ is preferred over word v^- for topic k is given by

$$P(v^+ \succ v^- | k) = \sigma(r(k, v^+) - r(k, v^-)) \quad (6)$$

where $r(k, v)$ denotes the reward associated with assigning word v to topic k . Given preference pairs

(v^+, v^-) , the reward model can be trained by maximizing the likelihood of the observed preferences, leading to the negative log-likelihood objective

$$\mathcal{L}_{\text{RM}}^k = -\mathbb{E}_{(k, v^+, v^-)} \log \sigma(r(k, v^+) - r(k, v^-)) \quad (7)$$

In RLHF, policy optimization is commonly formulated as maximizing expected reward while regularizing the learned policy to stay close to a reference policy. The KL-regularized objective is

$$\max_{\phi} \mathbb{E}_{v \sim \pi_{\phi}(v|k)} [r(k, v)] - \alpha D_{\text{KL}}[\pi_{\phi}(v|k) \parallel \pi_{\phi}^{\text{ref}}(v|k)] \quad (8)$$

Following (Rafailov et al., 2023), the optimal solution to Eq. 8 admits the closed-form expression

$$\pi_{\phi}^*(v|k) = \frac{1}{C(k)} \pi_{\phi}^{\text{ref}}(v|k) \exp\left(\frac{r(k, v)}{\alpha}\right), \quad (9)$$

$$C(k) = \sum_{v \in V} \pi_{\phi}^{\text{ref}}(v|k) \exp\left(\frac{r(k, v)}{\alpha}\right) \quad (10)$$

Taking the logarithm of both sides and rearranging terms yields

$$r(k, v) = \alpha \log \frac{\pi_{\phi}^*(v|k)}{\pi_{\phi}^{\text{ref}}(v|k)} + \alpha \log C(k).$$

Substituting this expression into Eq. 6, the partition term cancels when comparing two candidate words. The preference probability can therefore be written directly in terms of the optimal policy and the reference policy:

$$P(v^+ \succ v^- | k) = \sigma\left(\alpha \log \frac{\pi_\phi^*(v^+ | k)}{\pi_\phi^{\text{ref}}(v^+ | k)}\right) \quad (11)$$

$$- \alpha \log \frac{\pi_\phi^*(v^- | k)}{\pi_\phi^{\text{ref}}(v^- | k)} \quad (12)$$

This allows us to optimize a parametrized policy π_ϕ directly from preference data without explicitly learning a reward model. The corresponding maximum likelihood objective becomes

$$\begin{aligned} \mathcal{L}_{\text{pref}}^k = & -\mathbb{E}_{(k, v^+, v^-)} \log \sigma\left(\alpha \log \frac{\pi_\phi^*(v^+ | k)}{\pi_\phi^{\text{ref}}(v^+ | k)}\right. \\ & \left. - \alpha \log \frac{\pi_\phi^*(v^- | k)}{\pi_\phi^{\text{ref}}(v^- | k)}\right) \end{aligned} \quad (13)$$

Recall that $\pi_\phi(v | k) = \text{softmax}_v(\ell_{:,k})_v$, we have $\pi_\phi(v | k) = \frac{e^{\ell_{vk}}}{Z_k}$ with $Z_k = \sum_{v \in V} e^{\ell_{vk}}$. So we have:

$$\begin{aligned} \log \frac{\pi_\phi^*(v | k)}{\pi_\phi^{\text{ref}}(v | k)} &= (\ell_{vk} - \log Z_k) \\ &= (\ell_{vk}^{\text{ref}} - \log Z_k^{\text{ref}}) \\ &= (\ell_{vk} - \ell_{vk}^{\text{ref}}) \\ &= (\log Z_k - \log Z_k^{\text{ref}}) \end{aligned} \quad (14)$$

Finally, substituting the above expression into Eq. 13 yields the DPO loss in Eq. 1, that:

$$\mathcal{L}_{\text{pref}}^k = -\mathbb{E}_{(k, v^+, v^-)} \log \sigma\left[\alpha((\ell_{v^+,k} - \ell_{v^-,k}) - (\ell_{v^+,k}^{\text{ref}} - \ell_{v^-,k}^{\text{ref}}))\right] \quad (15)$$

We also present the detailed training algorithm of LLMFeed in Algorithm 1.

D Experimental details

D.1 Datasets

Our experiments are conducted on four widely used benchmark datasets. The 20 Newsgroups (20NG) dataset (Lang, 1995) contains news documents categorized into 20 topics. Web of Science

Dataset	# of texts	average text length	# of labels	vocab size
20NG	18,846	110.5	20	5,000
WoS	11,967	109.42	33	10,000
BBC	2,225	183.2	5	5,000
NYT	9,172	175.4	12	10,000

Table 9: Dataset statistics after preprocessing.

(WoS) (Kowsari et al., 2017) consists of scientific publications labeled across 33 research fields. The BBC dataset (Greene and Cunningham, 2006) includes articles published on the BBC news website during 2004-2005. Finally, the New York Times (NYT) dataset (Wu et al., 2024b) provides news articles grouped into 12 categories.

Before training the models, all datasets were processed to obtain bag-of-words (BoW) representations following the procedure described in (Wu et al., 2023). In particular, tokens that appeared fewer than three times in the corpus were removed, and documents containing fewer than two remaining tokens were excluded. These preprocessing steps were implemented using the TopMost toolkit². The resulting dataset statistics after preprocessing are summarized in Table 9.

D.2 Evaluation metrics

We evaluate topic quality using several widely adopted metrics, including C_v , NPMI, LLM score, Topic Diversity (TD), Harmonic Purity, Normalized Mutual Information (NMI), and Adjusted Rand Index (ARI). While previous works (Wu et al., 2023; Pham et al., 2024b; Yang et al., 2025) typically report only a single coherence metric (either C_v or NPMI), we report both metrics to provide a more comprehensive evaluation of LLMFeed’s performance. Furthermore, both coherence metrics (C_v and NPMI) are computed using a Wikipedia reference corpus rather than the training dataset. This setting provides a more reliable assessment of topic quality and better reflects the models’ generalization ability. For the LLM-based evaluation, we follow the protocol of Stambach et al. (2023) with GPT-5 as the evaluation LLM to ensure a fair and reliable assessment. Specifically, each topic is rated on a three-point scale: 1 (bad), 2 (medium), and 3 (good). We report the average score across all topics as the final LLM score.

²<https://github.com/bobxwu/topmost>

Method	20NG				WOS				BBC				NYT			
	C_v	LLMscore	NPMI	TD	C_v	LLMscore	NPMI	TD	C_v	LLMscore	NPMI	TD	C_v	LLMscore	NPMI	TD
ECRTM	0.500 ± 0.008	2.56 ± 0.04	0.053 ± 0.004	<u>0.984</u> ± 0.007	0.435 ± 0.005	2.64 ± 0.05	0.039 ± 0.005	0.999 ± 0.001	0.459 ± 0.005	2.84 ± 0.04	0.027 ± 0.003	0.963 ± 0.005	0.449 ± 0.002	2.72 ± 0.03	0.024 ± 0.002	0.999 ± 0.001
Neuromax	0.516 ± 0.012	<u>2.64</u> ± 0.05	<u>0.060</u> ± 0.003	0.995 ± 0.001	0.454 ± 0.005	2.64 ± 0.06	0.036 ± 0.004	0.999 ± 0.001	<u>0.479</u> ± 0.004	<u>2.86</u> ± 0.05	<u>0.035</u> ± 0.003	<u>0.969</u> ± 0.004	0.439 ± 0.014	2.68 ± 0.04	0.025 ± 0.004	0.999 ± 0.001
CGTM	0.461 ± 0.012	2.36 ± 0.04	-0.001 ± 0.002	0.771 ± 0.011	0.462 ± 0.008	2.70 ± 0.02	0.030 ± 0.005	0.792 ± 0.014	0.467 ± 0.008	2.60 ± 0.03	0.001 ± 0.002	0.747 ± 0.004	<u>0.465</u> ± 0.006	2.64 ± 0.03	0.003 ± 0.004	<u>0.891</u> ± 0.007
LLM-ITL	<u>0.522</u> ± 0.007	2.56 ± 0.06	0.057 ± 0.003	0.957 ± 0.005	<u>0.468</u> ± 0.007	<u>2.72</u> ± 0.04	<u>0.047</u> ± 0.003	0.994 ± 0.004	0.471 ± 0.006	<u>2.86</u> ± 0.04	0.033 ± 0.001	0.900 ± 0.009	<u>0.465</u> ± 0.002	<u>2.74</u> ± 0.05	<u>0.035</u> ± 0.004	0.999 ± 0.001
LLMFeed	0.547[‡] ± 0.005	2.76[‡] ± 0.04	0.073[‡] ± 0.003	<u>0.984</u> ± 0.006	0.493[‡] ± 0.008	2.80[‡] ± 0.03	0.065[‡] ± 0.003	0.999 ± 0.001	0.501[‡] ± 0.004	2.96[‡] ± 0.03	0.047[‡] ± 0.003	0.979[‡] ± 0.004	0.481[‡] ± 0.008	2.92[‡] ± 0.03	0.045[‡] ± 0.001	0.999 ± 0.001

Table 10: Comparison of topic quality across four datasets using four metrics at $K = 25$ topics. Each cell reports the mean value with standard deviation, computed over five runs with different random seeds. Best and second-best results are shown in bold and underline. [‡] indicates that LLMFeed outperforms the strongest baseline under a two-sided t-test with $p < 0.05$.

Method	20NG				WOS				BBC				NYT			
	C_v	LLMscore	NPMI	TD	C_v	LLMscore	NPMI	TD	C_v	LLMscore	NPMI	TD	C_v	LLMscore	NPMI	TD
ECRTM	0.485 ± 0.006	2.56 ± 0.04	0.044 ± 0.005	0.944 ± 0.014	0.450 ± 0.007	2.68 ± 0.05	0.033 ± 0.002	0.984 ± 0.014	0.448 ± 0.007	2.82 ± 0.06	0.028 ± 0.004	<u>0.956</u> ± 0.007	0.430 ± 0.007	2.70 ± 0.03	0.030 ± 0.002	0.999 ± 0.001
Neuromax	<u>0.501</u> ± 0.013	<u>2.62</u> ± 0.04	<u>0.051</u> ± 0.003	<u>0.957</u> ± 0.001	0.446 ± 0.011	2.68 ± 0.03	0.031 ± 0.002	0.970 ± 0.007	0.438 ± 0.005	2.82 ± 0.05	0.028 ± 0.001	0.949 ± 0.008	<u>0.446</u> ± 0.003	<u>2.78</u> ± 0.05	0.031 ± 0.001	0.999 ± 0.001
CGTM	0.424 ± 0.014	2.34 ± 0.03	-0.021 ± 0.003	0.785 ± 0.004	0.462 ± 0.009	<u>2.76</u> ± 0.06	0.026 ± 0.001	0.744 ± 0.009	0.433 ± 0.009	2.56 ± 0.04	-0.005 ± 0.002	0.685 ± 0.010	0.441 ± 0.012	2.72 ± 0.02	0.001 ± 0.004	0.821 ± 0.010
LLM-ITL	0.492 ± 0.007	<u>2.62</u> ± 0.02	0.048 ± 0.003	0.891 ± 0.012	<u>0.467</u> ± 0.011	<u>2.76</u> ± 0.04	<u>0.039</u> ± 0.002	<u>0.986</u> ± 0.009	<u>0.452</u> ± 0.005	<u>2.86</u> ± 0.02	<u>0.032</u> ± 0.003	0.917 ± 0.011	0.439 ± 0.008	<u>2.78</u> ± 0.02	<u>0.037</u> ± 0.001	0.994 ± 0.005
LLMFeed	0.522[‡] ± 0.010	2.74[‡] ± 0.03	0.059[‡] ± 0.003	0.973[‡] ± 0.007	0.478[‡] ± 0.008	2.86[‡] ± 0.04	0.045[‡] ± 0.003	0.989 ± 0.008	0.466[‡] ± 0.003	2.90[‡] ± 0.02	0.035 ± 0.003	0.965[‡] ± 0.005	0.453[‡] ± 0.005	2.84[‡] ± 0.03	0.044[‡] ± 0.004	0.997 ± 0.002

Table 11: Comparison of topic quality across four datasets using four metrics at $K = 50$ topics. Each cell reports the mean value with standard deviation, computed over five runs with different random seeds. Best and second-best results are shown in bold and underline. [‡] indicates that LLMFeed outperforms the strongest baseline under a two-sided t-test with $p < 0.05$.

D.3 LLMFeed Implementation Details

To ensure a fair comparison with prior methods (Dieng et al., 2020; Wu et al., 2023; Pham et al., 2024b), we adopt the same base VAE configuration as ECRTM (Wu et al., 2023). Specifically, the encoder is implemented as a fully connected neural network with two linear layers, each containing 200 hidden units. We initialize the word embeddings using pre-trained GloVe vectors. Following the findings of ECRTM, freezing the pre-trained word embeddings yields significantly more coherent topics; therefore, we adopt this setting in all experiments.

For the pre-trained language embedding model (PLEM) used to encode texts in the Preference Word Recommendation and Topic-guided Contrastive Learning components, we employ the BAAI General Embedding (BGE) model (Chen et al., 2024) (small version)³. The large language model (LLM) used in our framework is GPT-4o, with prompting details provided in Appendix B.

³BAAI/bge-small-en-v1.5

Our model is implemented in PyTorch⁴.

In terms of hardware, LLMFeed can be trained on a standard GPU such as an NVIDIA RTX 3060 (16GB). However, since certain baseline methods, such as LLM-ITL (Yang et al., 2025), require an NVIDIA A100 GPU with 80GB memory, we conduct all experiments on an NVIDIA A100 80GB GPU to ensure consistent experimental conditions.

Regarding hyper-parameters, the preference margin parameter α is set to 1 across all experiments. In Topic-guided Contrastive Learning, we select the top two topics for each document to construct the contrastive set. The temperature parameter γ for the contrastive loss is fixed at 0.5. To ensure fair comparison with other neural topic models (NTMs), which are trained for a total of 500 epochs, we set the number of epochs for Stage 1 to $E = 350$ and for Stage 3 to $E' = 150$.

The following hyper-parameters are tuned via grid search using C_v coherence as the tuning criterion, and the selected configuration is then used to

⁴<https://pytorch.org/>

Method	20NG				WOS				BBC				NYT			
	C_v	LLMscore	NPMI	TD	C_v	LLMscore	NPMI	TD	C_v	LLMscore	NPMI	TD	C_v	LLMscore	NPMI	TD
ECRTM	0.460 ± 0.009	2.58 ± 0.04	0.030 ± 0.003	0.875 ± 0.011	0.434 ± 0.006	2.73 ± 0.04	0.022 ± 0.003	0.947 ± 0.009	0.433 ± 0.007	2.83 ± 0.04	0.023 ± 0.003	0.886 ± 0.010	0.408 ± 0.008	2.72 ± 0.04	0.017 ± 0.003	0.961 ± 0.008
Neuromax	0.473 ± 0.010	2.62 ± 0.04	0.038 ± 0.003	<u>0.895</u> ± 0.006	0.441 ± 0.008	2.72 ± 0.05	0.028 ± 0.003	<u>0.956</u> ± 0.006	0.436 ± 0.006	2.80 ± 0.05	0.023 ± 0.002	0.914 ± 0.007	0.412 ± 0.009	<u>2.81</u> ± 0.04	<u>0.025</u> ± 0.002	0.984 ± 0.004
CGTM	0.406 ± 0.013	2.28 ± 0.04	-0.031 ± 0.004	0.691 ± 0.012	0.437 ± 0.009	2.74 ± 0.04	0.020 ± 0.003	0.651 ± 0.014	0.430 ± 0.010	2.64 ± 0.05	-0.009 ± 0.003	0.626 ± 0.013	<u>0.428</u> ± 0.010	2.65 ± 0.05	-0.002 ± 0.004	0.767 ± 0.013
LLM-ITL	<u>0.492</u> ± 0.007	<u>2.67</u> ± 0.03	<u>0.047</u> ± 0.002	<u>0.822</u> ± 0.010	0.458 ± 0.007	<u>2.77</u> ± 0.03	<u>0.038</u> ± 0.002	0.943 ± 0.009	<u>0.450</u> ± 0.006	<u>2.86</u> ± 0.03	<u>0.031</u> ± 0.002	0.888 ± 0.010	0.411 ± 0.008	2.80 ± 0.04	0.022 ± 0.003	0.945 ± 0.009
LLMFeed	0.496 [‡] ± 0.004	2.75 [‡] ± 0.02	0.049 [‡] ± 0.002	0.898 [‡] ± 0.004	<u>0.456</u> ± 0.006	2.83 [‡] ± 0.02	0.039 [‡] ± 0.002	0.977 [‡] ± 0.003	0.460 [‡] ± 0.004	2.91 [‡] ± 0.02	0.038 [‡] ± 0.001	<u>0.895</u> ± 0.006	0.444 [‡] ± 0.004	2.88 [‡] ± 0.02	0.031 [‡] ± 0.001	0.968 ± 0.005

Table 12: Comparison of topic quality across four datasets using four metrics at $K = 75$ topics. Each cell reports the mean value with standard deviation, computed over five runs with different random seeds. Best and second-best results are shown in bold and underline. ‡ indicates that LLMFeed outperforms the strongest baseline under a two-sided t-test with $p < 0.05$.

Method	20NG				WOS				BBC				NYT			
	C_v	LLMscore	NPMI	TD	C_v	LLMscore	NPMI	TD	C_v	LLMscore	NPMI	TD	C_v	LLMscore	NPMI	TD
ECRTM	0.463 ± 0.008	2.61 ± 0.04	0.032 ± 0.003	<u>0.857</u> ± 0.008	0.418 ± 0.009	2.70 ± 0.05	0.018 ± 0.003	0.910 ± 0.010	0.423 ± 0.007	2.78 ± 0.04	0.022 ± 0.003	<u>0.883</u> ± 0.007	0.403 ± 0.008	2.81 ± 0.04	0.019 ± 0.003	<u>0.958</u> ± 0.006
Neuromax	0.468 ± 0.010	2.63 ± 0.04	0.036 ± 0.003	0.823 ± 0.011	0.425 ± 0.010	<u>2.75</u> ± 0.04	0.028 ± 0.002	<u>0.936</u> ± 0.007	0.426 ± 0.008	<u>2.80</u> ± 0.04	0.021 ± 0.002	0.854 ± 0.009	0.407 ± 0.008	2.82 ± 0.04	0.020 ± 0.002	0.954 ± 0.007
CGTM	0.400 ± 0.014	2.34 ± 0.05	-0.033 ± 0.004	0.629 ± 0.013	0.418 ± 0.011	2.71 ± 0.05	0.010 ± 0.004	0.622 ± 0.014	0.422 ± 0.010	2.64 ± 0.05	-0.012 ± 0.004	0.532 ± 0.014	0.395 ± 0.011	2.61 ± 0.05	-0.009 ± 0.004	0.681 ± 0.014
LLM-ITL	<u>0.482</u> ± 0.007	<u>2.67</u> ± 0.03	<u>0.040</u> ± 0.002	<u>0.767</u> ± 0.010	<u>0.430</u> ± 0.008	<u>2.75</u> ± 0.03	<u>0.033</u> ± 0.002	0.911 ± 0.009	<u>0.432</u> ± 0.007	<u>2.80</u> ± 0.03	<u>0.030</u> ± 0.002	0.837 ± 0.010	<u>0.408</u> ± 0.007	<u>2.84</u> ± 0.03	<u>0.021</u> ± 0.002	0.951 ± 0.009
LLMFeed	0.496 [‡] ± 0.004	2.74 [‡] ± 0.02	0.047 [‡] ± 0.001	0.858 [‡] ± 0.006	0.440 [‡] ± 0.004	2.82 [‡] ± 0.02	0.037 [‡] ± 0.001	0.945 [‡] ± 0.004	0.438 [‡] ± 0.004	2.86 [‡] ± 0.02	0.034 [‡] ± 0.001	0.885 ± 0.006	0.415 [‡] ± 0.004	2.93 [‡] ± 0.02	0.028 [‡] ± 0.001	0.959 ± 0.005

Table 13: Comparison of topic quality across four datasets using four metrics at $K = 100$ topics. Each cell reports the mean value with standard deviation, computed over five runs with different random seeds. Best and second-best results are shown in bold and underline. ‡ indicates that LLMFeed outperforms the strongest baseline under a two-sided t-test with $p < 0.05$.

report all evaluation metrics.

- $\lambda_{\text{ECR}} \in \{50, 80, 100, 120, 150\}$
- $\lambda_{\text{pref}} \in \{10, 20, 30, 40, 50\}$
- $\lambda_{\text{con}} \in \{10, 20, 30, 40, 50\}$

D.4 Baselines

We describe the baseline models used in our experiments as follows:

- LDA (Blei et al., 2003): A probabilistic topic model that applies Dirichlet priors to the document-topic and topic-word distributions.
- ETM (Dieng et al., 2020): A topic model that captures topic-word relationships through the dot product between word and topic embeddings.
- CTM (Bianchi et al., 2021a): A contextualized topic model that jointly leverages Bag-of-Words representations and contextualized document embeddings.
- ECRTM (Wu et al., 2023): A topic model that introduces embedding clustering regularization to improve topic diversity.
- Bertopic (Grootendorst, 2022): A clustering-based topic modeling method that groups document embeddings and employs a class-based TF-IDF variant to construct topic representations.
- Ctop2vec (Angelov and Inkpen, 2024): A clustering-based topic model that leverages sub-document and phrase-level semantic representations.
- CAST (Ma et al., 2025): An enhanced clustering-based model that filters less informative tokens using self-similarity.
- FASTopic (Wu et al., 2024b): A topic model that jointly captures semantic relationships among documents, topics, and words using optimal transport distance.

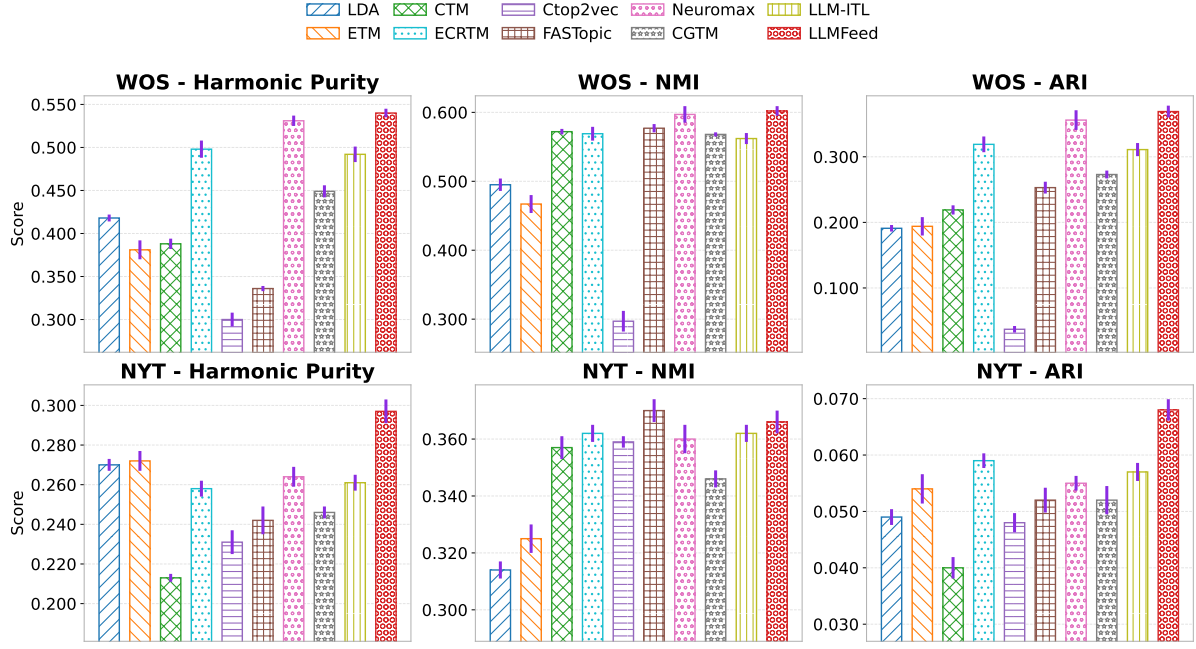


Figure 3: Document-topic alignment results measured by Harmonic Purity, NMI and ARI on WOS and NYT datasets. Results are mean of five runs; purple bars indicate std.

- Neuromax (Pham et al., 2024b): A model that improves document-topic distributions by maximizing mutual information between documents using pre-trained document embeddings.
- CGTM (Liu et al., 2025): A model that integrates contextual embeddings with graph information to enhance topic modeling performance.
- LLM-ITL (Yang et al., 2025): A method that refines topic words through an LLM-in-the-loop framework combined with optimal transport.
- TopicGPT (Pham et al., 2024a): A model that extracts topics from a corpus using prompt-based instructions with large language models.

For fair comparison, all methods are trained on an NVIDIA A100 80GB GPU, although some models (e.g., ETM and ECRTM) can be trained using standard GPUs. For LLM-ITL, to ensure comparability with our framework, we adopt ECRTM as the base model and freeze the word embeddings to maximize ECRTM’s performance. In the original LLM-ITL experiments⁵, the word embeddings

⁵<https://github.com/Xiaohao-Yang/LLM-ITL.git>

are trainable; however, this setting significantly degrades the performance of ECRTM. Following the configurations described in the original papers, we use LLaMA3 (Grattafiori et al., 2024) for LLM-ITL and GPT-4 (OpenAI et al., 2024a) for TopicGPT.

E Additional results

E.1 Topic quality across different number of topics

Tables 10-13 provide detailed topic-quality results under different numbers of topics ($K = 25, 50, 75, 100$), complementing the averaged results reported in the main paper. These results further show that the effectiveness of LLMFeed is not limited to a particular topic-number setting, but remains stable across both smaller and larger topic spaces.

For topic coherence, LLMFeed consistently achieves the strongest results across most datasets and values of K , as reflected by higher C_v , LLM-score, and NPMI. This suggests that LLMFeed improves the semantic quality of topic words in a robust manner. The improvement can be attributed to its LLM-driven feedback mechanism: after the base topic model learns stable topics, the LLM evaluates each topic, identifies its semantic meaning, and recommends more suitable topic words by separating relevant words from noisy or overly generic ones. These feedback signals are then converted

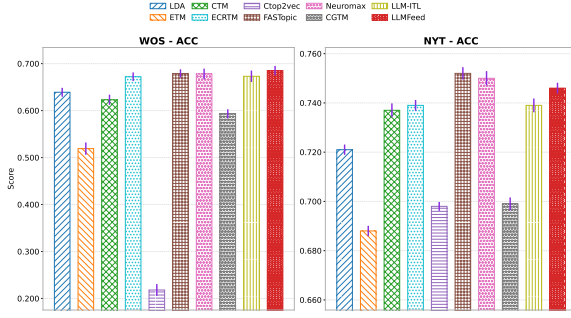


Figure 4: Document-topic alignment results measured by Accuracy on WOS and NYT datasets. Results are mean of five runs; purple bars indicate std.

Method	20NG			NYT		
	Purity	NMI	ARI	Purity	NMI	ARI
ECRTM	0.486	0.448	0.297	0.393	0.320	0.117
LLM-ITL	0.480	0.441	0.292	0.387	0.318	0.104
TopicGPT	0.417	0.358	0.197	0.364	0.230	0.104
LLMFeed	0.504	0.479	0.311	0.410	0.331	0.130

Table 14: Additional quantitative comparison with TopicGPT on document-topic assignment. TopicGPT produces an uncontrolled number of topics, while ECRTM, LLM-ITL, and LLMFeed are evaluated with $K = 25$ topics.

into preference pairs, allowing DPO to promote words that better match the topic meaning while suppressing less relevant alternatives. As a result, LLMFeed improves coherence not merely by re-ranking existing top words, but by directly guiding the topic-word distribution toward more semantically meaningful representations.

For topic diversity, LLMFeed also remains competitive across different topic-number settings. Although the gains in TD are generally less pronounced than those in coherence metrics, LLMFeed obtains the best or near-best TD in most cases, indicating that the coherence improvement does not come at the cost of severe topic overlap. This is important because TD complements coherence metrics: high C_v or NPMI is more meaningful when TD remains high, since it shows that the model captures multiple semantic aspects of the corpus rather than concentrating quality on a small set of redundant topics. In contrast, LLM-ITL often improves coherence over ECRTM but reduces TD, suggesting that its refinement may make topics semantically closer. LLMFeed mitigates this issue through the topic meaning identification step, where the LLM is instructed to avoid semantic overlap across topics, and through the reference-policy regularization in DPO, which prevents the refined topics from

drifting too far from the base topic structure. As K increases, TD naturally tends to decrease because more topics must be separated within the same semantic space; nevertheless, values around 0.85-0.90 already indicate that topic collapse is not severe (Pham et al., 2024b; Liu et al., 2025). Overall, LLMFeed maintains a favorable balance between coherent topic-word representations and sufficient topic-level diversity across different choices of K .

E.2 Additional Results on Document-Topic Distribution Quality

In this subsection, we further examine document-topic alignment on the WOS and NYT datasets with 100 topics, complementing the results on 20NG and BBC reported in the main paper (Subsection 4.3). Detailed results are presented in Figure 3. Overall, the trends are consistent with our main findings: **LLMFeed** achieves the strongest or near-strongest performance across most settings and remains particularly robust on clustering-oriented alignment metrics (Harmonic Purity, NMI, ARI). While several embedding-regularized baselines (e.g., FASTopic, Neuromax, and CGTM) are competitive on specific metrics, their performance is less consistent across datasets. In contrast, LLMFeed maintains stable and high-quality alignment behavior, indicating that its improvements are systematic rather than case-specific.

In addition to Harmonic Purity, NMI, and ARI, we also compare the methods in terms of classification accuracy on the WOS and NYT datasets. Specifically, we use the document-topic distributions from the training set to train SVM classifiers, which are then evaluated on the test sets. LLMFeed and models that integrate PLEM in the encoder, such as FASTopic and Neuromax, continue to demonstrate strong classification performance. Among them, LLMFeed achieves the best results on WOS, further confirming the effectiveness of topic-guided contrastive learning.

E.3 Additional Quantitative Comparison with TopicGPT

Since TopicGPT does not follow the standard top-word topic representation, prior work, including TopicGPT and LLM-ITL, mainly compares its topic representations qualitatively. To provide additional quantitative evidence, we report document-topic assignment results for TopicGPT on 20NG and NYT. Because TopicGPT produces an uncontrolled and often much smaller number of topics,

λ_{pref}	C_v	TD	Harmonic Purity	ARI
10	0.516	0.946	0.538	0.345
20	0.521	0.960	0.532	0.341
30	0.522	0.973	0.525	0.339
40	0.513	0.962	0.522	0.339
50	0.511	0.968	0.522	0.338

Table 15: Sensitivity analysis of LLMFeed with respect to the hyper-parameter λ_{pref} on the 20NG dataset with 50 topics.

λ_{con}	C_v	TD	Harmonic Purity	ARI
10	0.520	0.957	0.504	0.328
20	0.522	0.973	0.525	0.339
30	0.521	0.972	0.529	0.328
40	0.520	0.973	0.535	0.314
50	0.520	0.970	0.543	0.300

Table 16: Sensitivity analysis of LLMFeed with respect to the hyper-parameter λ_{con} on the 20NG dataset with 50 topics.

e.g., 9 topics on 20NG and 18 topics on NYT, direct comparison is not fully balanced. We therefore compare TopicGPT with ECRTM, LLM-ITL, and LLMFeed under the $K = 25$ setting for the latter three methods.

Table 14 shows that TopicGPT underperforms the NTM-based methods on document-topic assignment, especially on 20NG, where its Purity, NMI, and ARI are all substantially lower than ECRTM and LLM-ITL. This suggests that although free-form LLM generation can produce readable topic descriptions, it may capture broader corpus-level concepts and provide less fine-grained document-topic structure. The gap is also visible on NYT, particularly in NMI, where TopicGPT is clearly below the other methods. In contrast, LLMFeed achieves the best results on both datasets, indicating that grounding LLM feedback in corpus vocabulary and using it to refine an NTM preserves the fine-grained topic structure while improving semantic quality.

E.4 Qualitative Complexity Analysis

This appendix provides additional qualitative complexity analysis to complement the quantitative results in Sub-section 4.5. Let K be the number of topics, V the vocabulary size, and L the dimension of the pre-trained word embeddings.

For LLMFeed, excluding the LLM inference stage, the extra training cost beyond the ECRTM baseline comes from the DPO and contrastive refinement objectives. The DPO term requires an

additional pass over the topic-word structure to compute topic-word logits, leading to an overhead of $O(KVL)$. Since the number of preference candidates per topic is capped at a small constant (30 in our implementation), the pairwise comparison step remains linear in K and does not dominate the overall cost. The contrastive term further adds a batch-wise overhead of $O(KB^2)$, where B is the batch size. Overall, the additional training cost of LLMFeed is $O(KB^2 + KVL)$ compared with ECRTM.

By comparison, Neuromax (Pham et al., 2024b) introduces additional cost through its InfoNCE loss and Group Topic Regularization. Although the dominant terms differ, its practical overhead is qualitatively similar to that of LLMFeed: LLMFeed is mainly driven by the large vocabulary term KVL , whereas Neuromax is more influenced by the Sinkhorn-related term $T_{\text{gr}}K^2$ in Group Topic Regularization, where T_{gr} denotes the number of Sinkhorn iterations and is set relatively high to ensure convergence⁶. In practice, however, the additional LLM phase introduces some extra runtime for LLMFeed compared with Neuromax, as shown in Table 4.

In contrast, the main overhead of LLM-ITL (Yang et al., 2025) does not come from its OT regularization, whose complexity is only $O(K \cdot OT(m, s))$, where m is the number of top words extracted from the topic model and s is the number of words suggested by the LLM. Its main cost instead comes from frequent LLM calls, with one call made per batch. Even when we parallelize all 100 topics in Table 4.5, the total number of LLM calls still exceeds 2000. TopicGPT (Pham et al., 2024a) is even more expensive, as it queries the LLM at the document level, resulting in both high runtime and a large number of LLM calls. Although it provides a sampling-based variant using 1000 randomly selected documents, it still requires more than two hours to run.

E.5 Sensitivity Analysis

Sensitivity Analysis of Weight Parameters. In this paragraph, we analyze the sensitivity of LLMFeed with respect to the hyper-parameters λ_{pref} and λ_{con} , as reported in Tables 15 and 16, respectively. Overall, the configuration $\lambda_{\text{pref}} = 30$ and $\lambda_{\text{con}} = 20$ achieves the best balance between topic quality (C_v , TD) and document-topic alignment

⁶<https://github.com/Fsoft-AIC/NeuroMax.git>

Dataset	Embedding Model	C_v	NPMI	LLMscore	TD	HP	NMI	ARI
20NG	all-MiniLM-L6-v2	0.521	0.059	2.74	0.960	0.517	0.502	0.333
	all-mpnet-base-v2	0.520	0.059	2.74	0.960	0.515	0.499	0.339
	bge-small-en-v1.5	0.522	0.059	2.74	0.960	0.525	0.507	0.339

Table 17: Topic quality and clustering performance on 20NG (50 topics) using different embedding models. HP denotes Harmonic Purity.

(Harmonic Purity, ARI).

For λ_{pref} , a smaller value (e.g., $\lambda_{\text{pref}} = 10$) improves Harmonic Purity and ARI, but leads to noticeable decreases in topic coherence (C_v) and topic diversity (TD). Increasing λ_{pref} beyond the optimal value (e.g., 40 or 50) results in performance degradation in both topic quality and document-topic alignment.

Regarding λ_{con} , a small value ($\lambda_{\text{con}} = 10$) shows limited effectiveness in improving Harmonic Purity and ARI compared to other settings. Conversely, increasing λ_{con} beyond 20 gradually reduces topic quality and ARI, indicating that excessively strong contrastive regularization may negatively affect the model’s ability to maintain coherent and well-aligned topics.

Comparison of Different Pre-trained Language Embedding Models (PLEM). To investigate the impact of the pre-trained language embedding model (PLEM) used in our framework, we compare **bge-small-en-v1.5**, which is the default embedding model adopted in LLMFeed, with two widely used sentence embedding models: **all-MiniLM-L6-v2**⁷ and **all-mpnet-base-v2**⁸. These embeddings are used to represent documents in the *topic-guided contrastive learning* component.

As shown in Table 17, the performance differences across embedding models are relatively small, indicating that LLMFeed is generally robust to the choice of PLEM. Nevertheless, **bge-small-en-v1.5**, which we adopt in our model, achieves slightly better results on several metrics, including C_v , Harmonic Purity, and NMI. This observation supports our choice of **bge-small-en-v1.5** as the default embedding model, while also demonstrating that the proposed topic-guided contrastive learning framework can effectively operate with other commonly used sentence embedding models.

⁷[sentence-transformers/all-MiniLM-L6-v2](#)

⁸[sentence-transformers/all-mpnet-base-v2](#)

E.6 Ablation on LLM Feedback Design

To further examine the design choices in LLMFeed, we conduct additional ablations on the NYT dataset with $K = 50$ topics. We focus on two aspects: (i) the preference-selection rules used to construct win/lose word pairs from LLM feedback, and (ii) the epoch split between the initial base-topic training stage and the subsequent refinement stage. The results are reported in Table 18.

The rule ablation shows that the full instruction design achieves the best overall balance across topic quality and document-topic assignment metrics. Removing any individual rule still improves over ECRTM and LLM-ITL, which suggests that LLM feedback itself provides a strong semantic signal. However, these variants are slightly weaker than the full LLMFeed design. In particular, removing Rule 2 weakens the use of relevant words outside the current top-15 list, reducing the ability of LLMFeed to replace noisy or generic topic words with better candidates. Removing Rule 4 also leads to slightly lower performance, indicating that balanced win/lose pairs help construct more stable preference data for DPO training.

The effects of Rules 1 and 3 are comparatively smaller but still meaningful. Rule 1 ensures that clearly unrelated words among the top topic words are explicitly penalized, while Rule 3 preserves already good top words instead of unnecessarily replacing them. Together, these rules make the preference data less arbitrary: the LLM is not simply asked to rewrite topics freely, but is guided to identify which words should be promoted or suppressed according to the inferred topic meaning. This is consistent with the goal of LLMFeed, which is to refine the topic-word distribution while preserving the useful structure learned by the base topic model.

For the epoch ablation, we keep $E + E' = 500$ for fair comparison with other neural topic models trained for the same total number of epochs. The results show that LLMFeed performs best when the base topic model is first trained for around 300–

Concept	Model / Config	NYT- C_v	NYT-LLM	NYT-TD	NYT-HP	NYT-ARI
Baseline	ECRTM	0.430	2.70	0.999	0.325	0.088
	LLM-ITL	0.439	2.78	0.994	0.324	0.088
Rule	w/o Rule 1	0.451	2.80	0.999	0.340	0.094
	w/o Rule 2	0.449	2.80	0.997	0.339	0.093
	w/o Rule 3	0.452	2.84	0.997	0.340	0.091
	w/o Rule 4	0.450	2.80	0.996	0.339	0.092
	LLMFeed	0.453	2.84	0.997	0.340	0.094
Epoch	(200/300)	0.442	2.80	0.997	0.330	0.088
	(250/250)	0.448	2.82	0.995	0.332	0.090
	(300/200)	0.453	2.84	0.996	0.339	0.094
	(350/150)	0.453	2.84	0.997	0.340	0.094
	(400/100)	0.452	2.84	0.997	0.340	0.094
	(450/50)	0.449	2.82	0.992	0.337	0.091

Table 18: Ablation on preference-selection rules and training epoch splits on the NYT dataset with $K = 50$ topics. “w/o” denotes removing the corresponding rule. Rule 1 requires bad words in the top-15 to be included in the lose list; Rule 2 requires good words outside the top-15 to be included in the win list; Rule 3 keeps good words inside the top-15 in the win list; and Rule 4 keeps the win/lose counts balanced. The notation (E/E') denotes the number of epochs for initial base-topic training and subsequent refinement, respectively. HP denotes Harmonic Purity.

Method	C_v	LLM score	TD	Harm. Purity	ARI
ECRTM	0.430	2.70	0.999	0.325	0.088
LLM-ITL	0.439	2.78	0.994	0.324	0.088
DF/TF-IDF $_{r=5}$	0.433	2.74	0.999	0.338	0.091
DF/TF-IDF $_{r=7}$	0.435	2.76	0.997	0.339	0.092
BGE-only $_{r=5}$	0.445	2.80	0.997	0.340	0.093
BGE-only $_{r=7}$	0.444	2.80	0.997	0.340	0.094
LLMFeed	0.453	2.84	0.997	0.340	0.094

Table 19: Comparison with simpler refinement baselines on the NYT dataset with $K = 50$ topics. DF/TF-IDF and BGE-only use cheaper proxy signals to construct win/lose word pairs while keeping the same candidate pool and DPO refinement stage as LLMFeed. The subscript r denotes the number of promoted/demoted words.

400 epochs before applying LLM-guided refinement. When E is too small, the base topics are not sufficiently stable, making the LLM-generated topic meanings and preference pairs less reliable. In contrast, when E is too large, the refinement stage becomes too short to fully exploit DPO and contrastive learning. The selected 350/150 split therefore provides a good trade-off between stable base-topic learning and effective LLM-guided refinement.

E.7 Comparison with Simpler Refinement Baselines

We further compare LLMFeed with two simpler refinement baselines that do not use LLM preference labeling. These baselines follow the same

overall refinement pipeline as LLMFeed: they use the same candidate pool, composed of the top-25 topic words and 5 additional words retrieved by embedding similarity, and apply the same subsequent DPO update. The only difference is how the win/lose word pairs are constructed. This allows us to isolate the effect of LLM-based semantic preference labeling from cheaper proxy signals.

The first baseline, *DF/TF-IDF refinement*, selects lose words from the current top-15 topic words based on high document frequency, treating them as generic or stopword-like terms. It then selects win words from outside the top-15 based on topic-specific TF-IDF scores computed from pseudo-documents assigned to the topic. The second baseline, *BGE-only refinement*, removes the LLM entirely and uses embedding similarity instead: it computes cosine similarity between each candidate word embedding and the mean embedding of the current top-15 topic words, then demotes the least similar top-15 words and promotes the most similar outside candidates. We report results with $r \in \{5, 7\}$, where r denotes the number of promoted/demoted words.

Table 19 shows that both proxy baselines improve over the ECRTM base model, confirming that cheaper signals such as generic-word removal and embedding similarity can provide useful refinement cues. However, DF/TF-IDF refinement remains relatively weak. Although it slightly improves over ECRTM, it still underperforms LLM-ITL on coherence metrics, suggesting that simply

Method	20NG				NYT			
	C_v	NPMI	Purity	ARI	C_v	NPMI	Purity	ARI
ECRTM	0.485	0.044	0.471	0.297	0.430	0.030	0.345	0.093
LLMFeed (Llama 3)	0.514	0.052	0.523	0.337	0.447	0.039	0.358	0.102
LLMFeed (GPT-4o)	0.522	0.059	0.525	0.339	0.453	0.044	0.357	0.102

Table 20: Effect of the LLM choice in LLMFeed on 20NG and NYT with 50 topics.

Method	20NG				NYT			
	C_v	NPMI	Purity	ARI	C_v	NPMI	Purity	ARI
ProdLDA	0.353	-0.065	0.492	0.303	0.334	-0.083	0.312	0.066
LLMFeed-ProdLDA	0.379	-0.033	0.508	0.322	0.377	-0.029	0.323	0.074
ETM	0.393	-0.027	0.370	0.224	0.399	-0.019	0.317	0.064
LLMFeed-ETM	0.425	-0.011	0.443	0.248	0.419	-0.007	0.336	0.072

Table 21: Performance of LLMFeed when instantiated with ProdLDA and ETM as alternative base topic models on 20NG and NYT with 50 topics. Bold values indicate improvements over the corresponding base model.

removing frequent words and promoting high TF-IDF candidates cannot fully capture topic-specific semantic appropriateness.

BGE-only refinement performs more strongly than DF/TF-IDF, indicating that embedding similarity is a useful proxy for identifying semantically related topic words. Nevertheless, it still trails LLMFeed on C_v and LLMscore. This gap suggests that embedding similarity can capture broad semantic relatedness, but it is less effective than LLM feedback at making topic-specific judgments, such as distinguishing genuinely representative words from words that are merely close in embedding space. In contrast, LLMFeed evaluates the topic meaning explicitly and constructs preference pairs according to whether candidate words support that meaning.

The document-topic assignment metrics show smaller differences across these refinement variants. Harmonic Purity and ARI improve over ECRTM for most refinement baselines, but the gap between BGE-only and LLMFeed is modest. This is expected because these simpler baselines mainly modify the topic-word refinement signal, while document-topic assignment in LLMFeed is primarily improved by the contrastive doc-topic component, which is kept unchanged in this comparison. Overall, the results show that simple proxy signals are helpful, but LLM-based preference labeling provides stronger and more reliable topic-coherence improvements.

E.8 Effect of the LLM Choice

To examine whether the effectiveness of LLMFeed depends on the choice of LLM, we additionally replace GPT-4o with Llama 3 while keeping the remaining configuration unchanged. Table 20 reports the results on 20NG and NYT with 50 topics.

LLMFeed with Llama 3 consistently improves over the ECRTM base model across all metrics on both datasets, and its performance remains close to that of GPT-4o. For example, on 20NG, replacing GPT-4o with Llama 3 still improves C_v from 0.485 to 0.514 and Purity from 0.471 to 0.523. These results indicate that the improvements of LLMFeed are not tied to a particular proprietary LLM.

The small performance gap between Llama 3 and GPT-4o may partly arise from differences in output handling. With GPT-4o, we use function calling to enforce a predefined structured output, whereas the Llama 3 implementation relies on a textual output template followed by rule-based parsing. The latter may occasionally fail to recover all generated information, slightly affecting the final results.

E.9 Generalization to Different Base Topic Models

To investigate whether LLMFeed is specific to ECRTM, we further instantiate the framework with two additional neural topic models, ProdLDA and ETM. Table 21 presents the results on 20NG and NYT with 50 topics.

LLMFeed consistently improves both ProdLDA and ETM across all evaluated metrics and datasets. On 20NG, for example, LLMFeed increases the

C_v score of ProdLDA from 0.353 to 0.379 and that of ETM from 0.393 to 0.425. Similar improvements are observed on NYT, together with consistent gains in NPMI, Purity, and ARI.

These results demonstrate that LLMFeed is not specific to ECRTM and can be effectively integrated with different neural topic modeling architectures, supporting the generalizability of the proposed framework.